NAT'L INST OF STAND & TECH R.I.C.
A 59342
NIST
PUBLICATIONS

ANSI X3.23b-1993

ADOPTED FOR USE BY THE FEDERAL GOVERNMENT

FIPS

PUB 21-4B

SEE NOTICE ON INSIDE

American National Standard

for Information Systems – Programming Language – Correction Amendment for COBOL

JK
468
.A8A3
NO.21-4B
1994

ANSI American National Standards Institute
11 West 42nd Street
New York, New York
10036

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

NOTICE ON FORMAT AND STYLE: The format and style of this document differs from ANSI style as specified in the *ANSI Style Manual for Preparation of Proposed American National Standards*. This style and format exception has been permitted to ensure technical accuracy.

Published by
American National Standards Institute
11 West 42nd Street, New York, New York 10036

Copyright ©1994 by American National Standards Institute
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission of the publisher.

Printed in the United States of America

APS2 C594/50

ANSI®
X3.23b-1993
Amendment to
ANSI X3.23-1985

American National Standard
for Information Systems –

Programming Language – Correction Amendment for COBOL

Secretariat

Computer and Business Equipment Manufacturers Association

Approved January 31, 1994

American National Standards Institute, Inc.

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 21-4, COBOL. For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator, National Institute of Standardsd Technology, Gaithersburg, MD 20899.

ACKNOWLEDGMENT

Any organization interested in reproducing the COBOL standard and specifications in whole or in part, using ideas from this document as the basis for an instruction manual or for any other purpose, is free to do so. However, all such organizations are requested to reproduce the following acknowledgment paragraphs in their entirety as part of the preface to any such publication (any organization using a short passage from this document, such as in a book review, is requested to mention "COBOL" in acknowledgment of the source, but need not quote the acknowledgment):

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the CODASYL COBOL Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

The authors and copyright holders of the copyrighted materials used herein

FLOW-MATIC (trademark of Sperry Rand Corporation), Programming for the UNIVAC (R) I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translater Form No. F 28-8013, copyrighted 1959 by IBM; FACT, DSI 27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell

have specifically authorized the use of this material in whole or in part, in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

Foreword (This foreword is not part of American National Standard X3.23b-1993.)

This amendment was prepared by the X3J4 COBOL Technical Committee of Accredited Standards Committee X3. The X3J4 COBOL Technical Committee considered ambiguities and errors identified by October 17, 1990. Where a solution required extensive change, the X3J4 COBOL Technical Committee deferred correction to a future standard.

Suggestions for improvement of this standard will be welcome. They should be sent to the Computer and Business Equipment Manufacturers Association, 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by the Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

James Converse, Chair
Donald C. Loughry, Vice-Chair
Joanne Flanagan, Secretary

Organization Represented	*Name of Representative*
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt.)
AMP, Inc.	Edward Kelly
	Charles Brill (Alt.)
Apple Computer, Inc.	Karen Higginbottom
Association of the Institute for Certification of Computer Professionals (AICCP)	Kenneth Zemrowski
AT&T/NCR	Thomas W. Kern
	Thomas F. Frost (Alt.)
Boeing Company	Catherine Howells
	Andrea Vanosdoll (Alt.)
Bull HN Information Systems, Inc.	William George
Compaq Computers	James Barnes
Digital Equipment Corporation	Delbert Shoemaker
	Kevin Lewis (Alt.)
Eastman Kodak Company	James Converse
	Michael Nier (Alt.)
Guide International, Inc.	Frank Kirshenbaum
	Harold Kuneke (Alt.)
Hewlett-Packard	Donald C. Loughry
Hitachi America, Ltd.	John Neumann
	Kei Yamashita (Alt.)
Hughes Aircraft Company	Harold Zebrack
IBM Corporation	Joel Urman
	Mary Anne Lawler (Alt.)
National Communications Systems	Dennis Bodson
Neville & Associates	Carlton Neville
National Institute of Standards and Technology	Robert E. Rountree
	Michael Hogan (Alt.)
Northern Telecom, Inc.	Mel Woinsky
	Subhash Patel (Alt.)
Recognition Technology Users Association	Herbert P. Schantz
	G. Edwin Hale (Alt.)
Share, Inc.	Gary Ainsworth
	David Thewlis (Alt.)
Sony Corporation	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel D. Cheatham (Alt.)
Sun Microsystems, Inc.	Scott Jameson
	Gary S. Robinson (Alt.)

Organization Represented	*Name of Representative*
3M Company	Eddie T. Morioka
	Paul D. Jahnke (Alt.)
Unisys Corporation	John Hill
	Stephen P. Oksala (Alt.)
U.S. Department of Defense	William Rinehuls
	C. J. Pasquariello (Alt.)
U.S. Department of Energy	Alton Cox
	Lawrence A. Wasson (Alt.)
U.S. General Services Administration	Douglas Arai
	Larry L. Jackson (Alt.)
Wintergreen Information Services	John Wheeler
Xerox Corporation	Dwight McBain
	Roy Pierce (Alt.)

Preface

This amendment, the second to the document entitled *American National Standard for Information Systems – Programming Language – COBOL,* ANSI X3.23-1985, ISO 1989:1985, presents correction of errors and clarification of ambiguities in *Programming Language COBOL.*

The first amendment to ANSI X3.23-1985, ISO 1989:1985, entitled, *American National Standard for Information Systems – Programming Language – Intrinsic Function Module for COBOL*, ANSI X3.23a-1989 and *Programming Languages – COBOL – Amendment 1: Intrinsic Function Module*, ISO 1989/Amendment 1, must be applied to ANSI X3.23-1985 (or ISO 1989:1985) before applying the corrections identified in this second amendment.

Within this document:

- "First Standard COBOL" refers to ANSI X3.23-1968, ISO R-1989:1972.
- "Second Standard COBOL" refers to ANSI X3.23-1974, ISO 1989:1978, which superseded first Standard COBOL.
- "Third Standard COBOL" refers to ANSI COBOL X3.23-1985, ISO 1989:1985, which superseded second Standard COBOL.
- Both "Amendment 1" and "Intrinsic Function Amendment" refer to ANSI X3.23a-1989, ISO 1989:1985/Amd.1:1992.
- Both "Amendment 2" and "Correction Amendment" refer to ANSI X3.23b-1993, ISO 1989:1985/Amd.2:1993.
- "Amended third Standard COBOL" refers to third Standard COBOL updated first with the content of amendment 1 and then with the content of amendment 2. The terms "Standard COBOL" and "COBOL" also refer to amended third Standard COBOL.

American National Standard
for Information Systems –

Programming Language – Correction Amendment for COBOL

AMENDMENT 2

The changes in the column entitled *Description of Change* within this amendment are to be applied to ANSI X3.23-1985 (or ISO 1989:1985), updated with ANSI X3.23a-1989 (or ISO 1989:1985/Amd.1:1992). The beginning page number for the change is identified in the column entitled **Page No.** :

- without parentheses for a page in ANSI X3.23-1985 (or ISO 1989:1985);
- within parentheses for a page in ANSI X3.23a-1989 (or ISO 1989:1985/Amd.1:1992).

When the column entitled **Page No.** lists both a page number without parentheses and a page number within parentheses, the text in ANSI X3.23-1985 (or ISO 1989:1985) must be updated by ANSI X3.23a-1989 (or ISO 1989:1985/Amd.1:1992) before applying the change specified in this amendment. When the column entitled **Page No.** lists only a page number within parentheses, then new text from ANSI X3.23a-1989 (or ISO 1989:1985/Amd.1:1992) should first be inserted into the COBOL Standard and then updated as directed within this document. Within this amendment, the changes must be made in the order listed. Shading is used in this amendment to indicate specifications unique to level 2 (boxing was used for this purpose in third Standard COBOL).

Page No.	Description of Change
Preface	Replace with preface from page iv of this document.
x	Add the following after the last entry for the Segmentation module in the table of contents: **SECTION A: INTRINSIC FUNCTION MODULE** Chapter 1: Introduction to the Intrinsic Function Module 1.1 Purpose of Intrinsic Function Module . . . . . A-27 1.2 Function-Name A-27

Page No. **Description of Change**

Chapter 2: General Description

2.1 Function Definition and Returned Value A-27
2.2 Arguments A-28
2.3 Types of Functions A-29
2.4 Definitions of Functions A-29
2.5 ACOS A-33
2.6 ANNUITY A-34
2.7 ASIN A-35
2.8 ATAN A-36
2.9 CHAR A-37
2.10 COS A-38
2.11 CURRENT-DATE A-39
2.12 DATE-OF-INTEGER A-41
2.13 DAY-OF-INTEGER A-42
2.14 FACTORIAL A-43
2.15 INTEGER A-44
2.16 INTEGER-OF-DATE A-45
2.17 INTEGER-OF-DAY A-46
2.18 INTEGER-PART A-47
2.19 LENGTH A-48
2.20 LOG A-49
2.21 LOG10 ..... A-50
2.22 LOWER-CASE A-51
2.23 MAX A-52
2.24 MEAN A-53
2.25 MEDIAN A-54
2.26 MIDRANGE A-55
2.27 MIN A-56
2.28 MOD A-57
2.29 NUMVAL A-58
2.30 NUMVAL-C A-59
2.31 ORD A-60
2.32 ORD-MAX A-61
2.33 ORD-MIN A-62
2.34 PRESENT-VALUE A-63
2.35 RANDOM A-64
2.36 RANGE A-65
2.37 REM A-66
2.38 REVERSE A-67
2.39 SIN A-68
2.40 SQRT A-69
2.41 STANDARD-DEVIATION A-70
2.42 SUM A-71
2.43 TAN A-72
2.44 UPPER-CASE A-73
2.45 VARIANCE A-74
2.46 WHEN-COMPILED A-75

Page No.	Description of Change
xi (A-1)	Add the following after the entry for 3.5: 3.6 American National Standard COBOL 1985, Amendment 2 ..... XVII-13
xi (A-1)	Replace the entry for 4.4 with the following: 4.4 ISO 1989:1985/Amd.1:1992 ..... XVII-15 4.5 ISO 1989:1985/Amd.2:1994 ..... XVII-15
xii	Replace the entries for 2.1 and 2.2 with the following: 2.1 Substantive Changes Not Affecting Existing Programs (Third Standard COBOL) ..... XVII-42 2.1A Substantive Changes Not Affecting Existing Programs (Intrinsic Function Amendment) ..... XVII-50 2.1B Substantive Changes Not Affecting Existing Programs (Correction Amendment) ..... XVII-50 2.2 Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL) ..... XVII-51 2.2A Substantive Changes Potentially Affecting Existing Programs (Intrinsic Function Amendment) ..... XVII-80 2.2B Substantive Changes Potentially Affecting Existing Programs (Correction Amendment) XVII-80
I-8	Paragraph 1.5.2.5.2 entitled *Extension Language Elements*, last paragraph, add the following as the new last sentence: This warning mechanism is required to flag only extensions that are syntactically distinguishable.
I-8	Paragraph 1.5.2.5.3 entitled *Reserved Words*, change to read: An implementation of Standard COBOL must recognize as reserved words all of the COBOL reserved words occurring in the specifications of the seven required modules and the five optional modules. (See page IV-45, COBOL Reserved Words.)
I-12 (A-4)	Change the entry for "Reference-modifier" to "Reference modification".
I-42 (A-6)	Change the entry for "Reference-modifier" to "Reference modification".

Page No.	Description of Change
II-1	Paragraph 2.1 entitled *File Attributes*, change a portion of the last sentence from "sequence of the keys for indexed files, the blocking factor, the padding character" to "sequence of the keys for indexed files, the minimum and maximum physical record size, the padding character".
II-14	Paragraph 4.2 entitled *Initial Values of Tables*, change numbered item 3 at the top of page II-14 to read: (3) Every occurrence of an element in a table can be assigned the same initial value using the VALUE clause.
II-35 (A-7)	Paragraph 8 entitled *Intrinsic Function Facility*, second paragraph, replace the last sentence with the following: If, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, then the result of such a reference is undefined.
III-1	Paragraph 1 entitled *Introduction*, second paragraph, change the last sentence to read: "Complete specifications for elements defined in this section can be located in Sections IV through XVI and Section A of this document."
III-4	Definition of COBOL Character Set, NOTE 2, first sentence, change "they may be used in character-strings." to "they may be used in character-strings and text words.".
III-10	Definition of Fixed File Attributes, sixth line, change "the blocking factor" to "the minimum and maximum physical record size".
III-12 (A-9)	Definition of Integer, item 3, change to read: "An integer function."
III-12 (A-8)	Definition of Integer Function, change "A function whose category is numeric ..." to "A function whose class and category are numeric ...".
III-25	Definition of Text Word, item 3, add the following as a new last sentence: One or both of the bounding separators may be a pseudo-text delimiter.

Page No.	Description of Change
IV-4 (A-9)	Paragraph 4.2.1 entitled *Separators*, rule 4, change second sentence from: "Parentheses may appear only ..." to "Except in pseudo-text, parentheses may appear only ...".
IV-5	Paragraph 4.2.1 entitled *Separators*, change rule 5 to read: The punctuation character quotation mark is a separator. An opening quotation mark must be immediately preceded by a space, left parenthesis, or opening pseudo-text delimiter. A closing quotation mark, when paired with an opening quotation mark, must be immediately followed by one of the separators space, comma, semicolon, period, right parenthesis, or closing pseudo-text delimiter. Separators immediately preceding the opening quotation mark are not part of the opening separator quotation mark. Separators immediately following the closing quotation mark are not part of the closing separator quotation mark.
IV-5 (A-9)	Paragraph 4.2.2.1, entitled *COBOL Words*, replace the paragraph with the following: A COBOL word is a character-string of not more than 30 characters which forms a user-defined word, a system-name, a reserved word, or a function-name. Each character of a COBOL word that is not a special character word is selected from the set of letters, digits, and the hyphen; the hyphen may not appear as the first or last character in such words. Each lowercase letter is considered to be equivalent to its corresponding uppercase letter. Within a source program the following apply: (1) For all COBOL words excluding the words LENGTH, RANDOM, and SUM: a. Reserved words form disjoint sets with user-defined words, system-names, and function-names. b. Function-names, system-names, and user-defined words form intersecting sets. The same COBOL word may be used as a function-name, as a system-name, and as a user-defined word. The classification of a specific occurrence of such COBOL words is determined by the context of the clause or phrase in which it occurs.

Page No.	Description of Change
	(2) For the COBOL words LENGTH, RANDOM, and SUM: a. The reserved words LENGTH, RANDOM, and SUM form an identical set with the function-names LENGTH, RANDOM, and SUM. The same COBOL word LENGTH, RANDOM, or SUM may be used as a function-name and as a reserved word. The classification of a specific occurrence of such COBOL words is determined by the context in which it occurs. b. The COBOL words LENGTH, RANDOM, and SUM form disjoint sets with user-defined words and system-names. The COBOL words LENGTH, RANDOM, and SUM may not be used as a system-name or as a user-defined word regardless of context.
IV-9 (A-10)	Paragraph 4.2.2.1.4 entitled *Function-Names*, change the second sentence to read: "With the exception of the words LENGTH, RANDOM, and SUM, a word that is a function-name may appear in different contexts in a program as either a system-name or as a user-defined word."
IV-11	Paragraph 4.2.2.2.3 entitled *Figurative Constant Values*, second occurrence of a paragraph numbered (1), penultimate sentence, change in part from: " ... until it is equal to the number of character positions in the associated data item." to "... until the number of character positions remaining is equal either to 1 or to the number of character positions in the associated data item, whichever is greater.".
IV-22 (A-11)	Paragraph 4.3.8.3.1 change the title from *Purpose of a Function-Identifier* to *Function.*
IV-22 (A-12)	Paragraph 4.3.8.4, change the title from *Reference-Modifier* to *Reference Modification.*
IV-27	Paragraph 4.4.4 entitled *Explicit and Implicit Scope Terminators*, last paragraph, item 2, delete the last sentence that reads: "Examples of such phrases are ELSE, WHEN, NOT AT END, etc."

Page No.	Description of Change
IV-38	Paragraph 6.4.2.2.1 entitled *Definition of Compiler Directing Statement*, second sentence, change to read: "The compiler directing verbs are COPY, ENTER, REPLACE, and USE (see page XII-2, The COPY Statement; page VI-83, The ENTER Statement; page XII-6, The REPLACE Statement; and the USE statement on pages VII-50, VIII-35, IX-39, XIII-76, XIII-78, and XV-5).".
IV-43	Paragraph 7.2.5 entitled *Pseudo-Text*, change a portion of the first sentence from "The character-strings and separators comprising" to "The text words and the separator space comprising".
IV-45 (A-13)	Paragraph 8 entitled *COBOL Reserved Words*, change the first paragraph to read: "The following is a list of COBOL reserved words for the seven required modules and the five optional modules."
IV-45	Paragraph 8 entitled *COBOL Reserved Words*, add the word FUNCTION between the words FROM and GENERATE in the list of COBOL reserved words.
IV-46 (A-14)	Paragraph 8 entitled *COBOL Reserved Words*, delete the following text: The following is a list of COBOL reserved words for the optional Intrinsic Function module. FUNCTION
V-24	General format of the IF statement, delete the braces surrounding statement-1; delete the ellipsis following statement-1; delete the braces surrounding statement-2; delete the ellipsis following statement-2.
VI-15	Paragraph 4.5.4 of SPECIAL-NAMES paragraph, general rule 4d, second sentence, change "The collating sequence identified" to "The character code set and/or collating sequence identified".

Page No.	Description of Change
VI-16	Paragraph 4.5.4 of SPECIAL-NAMES paragraph, general rule 4d, subitem 4, replace the first sentence (Beginning "If the THROUGH phrase is specified") with the following new sentence: If the THROUGH phrase is specified, the contiguous characters in the native character set, beginning with the character specified by literal-1 and ending with the character specified by literal-2, are each assigned a successive ascending collating position in the collating sequence being specified and/or a successive ascending ordinal number in the character code set being specified.
VI-16	Paragraph 4.5.4 of SPECIAL-NAMES paragraph, general rule 4d, renumber subitems 4 and 5 to new subitems 5 and 6. Insert the following as the new subitem 4: 4) If a character code set is being specified, the implementor defines the ordinal number within the character code set being specified for each character within the native character set that is not specified by the literal-1 phrase.
VI-31	Paragraph 5.9.4 of the PICTURE clause, general rule 8, explanation of symbol "P", second paragraph, change subparagraph b to read: b. An elementary MOVE statement where the sending operand is numeric and its PICTURE character-string contains the symbol 'P'.
VI-31	Paragraph 5.9.4 of the PICTURE clause, general rule 8, explanation of the symbol "P", second paragraph, shade subparagraph c.
VI-34	Paragraph 5.9.5, editing rules of the PICTURE clause, editing rule 7, second paragraph, change last sentence from: "... this string of floating insertion characters may have the fixed insertion characters 'CR' and 'DB' immediately to the right of this string." to "... this string of floating insertion characters may have one of the editing sign control symbols '+', '–', 'CR', or 'DB' immediately to the right of this string."

Page No.	Description of Change

VI-37 Paragraph 5.9.6, PICTURE precedence chart, delete the symbol "X" at the following two intersections:

1. The first symbol (at the top of the table) is the right-most "P" under "Other Symbols"; the second symbol (at the side of the table) is the lower "{±}" for "Floating Insertion Symbols".

2. The first symbol (at the top of the table) is the right-most "P" under "Other Symbols"; the second symbol (at the side of the table) is the lower "cs" for "Floating Insertion Symbols".

VI-56 Paragraph 6.3.1.1.3 entitled *Comparisons Involving Index-Names and/or Index Data Items*, change the first sentence of subparagraph 2 to read:

An index-name and a numeric data item or numeric literal.

VI-57 (A-18) Paragraph 6.3.1.2 entitled *Class Condition*, replace the first paragraph after the general format beginning with "Identifier-1 must reference" and the second paragraph after the general format beginning with "When used, NOT and" with the following:

Identifier-1 must reference a data item whose usage is explicitly or implicitly DISPLAY. If identifier-1 is a function-identifier, it must reference an alphanumeric function. When the class condition does not include the word NOT and identifier-1 is a zero-length group item, the result of the class test is always false.

When used, NOT and the next key word specify one class condition that defines the class test to be executed for truth value; e.g. NOT NUMERIC is a truth test for determining that an operand is nonnumeric. When the class condition includes the word NOT and identifier-1 is a zero-length group item, the result of the class test is always true.

VI-90 Paragraph 6.16.2, general format of the IF statement, delete the braces surrounding statement-1; delete the ellipsis following statement-1; delete the braces surrounding statement-2; delete the ellipsis following statement-2.

VI-91 Paragraph 6.16.4 of the IF statement, general rule 2b, add the following as the new last sentence:

This has the effect of causing a transfer of control to an implicit CONTINUE statement immediately preceding the next separator period.

Page No.	Description of Change
VI-91	Paragraph 6.16.4 of the IF statement, general rule 2e, add the following as the new last sentence: This has the effect of causing a transfer of control to an implicit CONTINUE statement immediately preceding the next separator period.
VI-91	Paragraph 6.16.4 of the IF statement, general rule 3, second paragraph, replace with the following: IF statements within IF statements are considered matched IF, ELSE, and END-IF ordered combinations, proceeding from left to right. Thus, any ELSE encountered is matched with the nearest preceding IF that either has not been already matched with an ELSE or has not been implicitly or explicitly terminated. Any END-IF encountered is matched with the nearest preceding IF that has not been implicitly or explicitly terminated.
VI-95	Paragraph 6.18.3 of the INSPECT statement, change syntax rule 8 to read: (8) When the CHARACTERS phrase is used, literal-3 or the size of the data item referenced by identifier-5 must be one character in length.
VI-96	Paragraph 6.18.4 of the INSPECT statement, insert the following as a new general rule 1 and renumber old general rules 1 through 7 as new general rules 2 through 8; renumber old general rules 8 through 17 as new general rules 10 through 19. (1) For the purpose of determining its length, identifier-1 is treated as if it were a sending data item. (See page VI-26, The OCCURS Clause).
VI-96	Paragraph 6.18.4, new general rule 2, last line, change "general rules 5 through 7." to "general rules 6 through 8.".
VI-96	Paragraph 6.18.4, new general rule 3b, fourth line, change "(see general rule 2a)" to "(see general rule 3a)".
VI-96	Paragraph 6.18.4, new general rule 3c, fifth line, change "in general rule 2b" to "in general rule 3b".
VI-96	Paragraph 6.18.4, new general rule 4, first line, change "In general rules 5 through 17" to "in general rules 6 through 19".

Page No.	Description of Change
VI-97	Paragraph 6.18.4, new general rule 7a, subitem 2, change "general rules 10 and 13" to "general rules 12 and 15".
VI-97	Paragraph 6.18.4, new general rule 7a, subitem 3, change "defined in general rule 13" to "defined in general rule 15".
VI-97	Paragraph 6.18.4, new general rule 7c, second line, change "in general rules 10 and 13" to "in general rules 12 and 15".
VI-97	Paragraph 6.18.4, new general rule 7e, second line, change "paragraphs 6a through 6d" to "general rules 7a through 7d".
VI-97	Paragraph 6.18.4, new general rule 8, first line, change "in general rule 6" to "in general rule 7".
VI-97	Paragraph 6.18.4, new general rule 8a, third line, change "in general rule 6" to "in general rule 7".
VI-98	Paragraph 6.18.4, new general rule 8b, eighth line from end of paragraph, change "in general rule 6" to "in general rule 7".
VI-98	Paragraph 6.18.4, new general rule 8c, seventh line from end of paragraph, change "in general rule 6" to "in general rule 7".
VI-98	Paragraph 6.18.4 of the INSPECT statement, add the following as the new general rule 9 located immediately after the header FORMAT 1: (9) Identifier-1 is a sending data item.
VI-98	Paragraph 6.18.4, new general rule 12c, third line, change "in general rule 6e" to "in general rule 7e".
VI-99	Paragraph 6.18.4, new general rule 15a, second line, change "in general rule 6e to "in general rule 7e".

Page No.	Description of Change
VI-113	Paragraph 6.21.4 of the PERFORM statement, general rule 10d, change the fourth and fifth sentences to read: If index-name-1 is specified, the value of identifier-3, index-name-2, or literal-1 at the beginning of the execution of the PERFORM statement must correspond to an occurrence number of an element in the table associated with index-name-1. If index-name-3 is specified, the value of identifier-6, index-name-4, or literal-3 at the beginning of the execution of the PERFORM statement must correspond to an occurrence number of an element in the table associated with index-name-3.
VI-124	Paragraph 6.22.4 of the SEARCH statement, general rule 2b, add the following as the new last sentence: When control is passed to the next executable sentence, the effect is a transfer of control to an implicit CONTINUE statement immediately preceding the next separator period.
VI-124	Paragraph 6.22.4 of the SEARCH statement, general rule 4, add the following as the new last sentence: When control is passed to the next executable sentence, the effect is a transfer of control to an implicit CONTINUE statement immediately preceding the next separator period.
VII-5	Delete paragraph 1.3.7 entitled *The File Attribute Conflict Condition*; renumber paragraph 1.3.8 entitled *Special Register LINAGE-COUNTER* as new paragraph 1.3.7.
VII-32	Paragraph 3.8.4 of the RECORD clause, general rule 10, add the following as a new last paragraph: If the number of character positions in the logical record to be written is less than integer-2 or greater than integer-3, the output statement is unsuccessful and, except during execution of a RELEASE statement, the associated I-O status is set to a value indicating the cause of the condition. (See page VII-2, I-O Status.)
VII-32	Paragraph 3.8.4 of the RECORD clause, renumber general rules 13 and 14 as new general rules 14 and 15. Add the following as new general rule 13 after the header FORMAT 3: (13) It is implementor defined whether format 3 of the RECORD clause produces fixed length records or variable length records.

Page No.	Description of Change

VII-37 Paragraph 4.2.4 of the CLOSE statement in the Sequential I-O module, general rule 3, change paragraph F to read as follows:

F. Close Reel/Unit

Input Files and Input-Output Files (Reel/Unit Media):

1) If the current reel/unit is the last or only reel/unit for the file, there is no reel/unit swap, the current volume pointer remains unchanged, and the file position indicator is set to indicate that no next reel/unit exists.

2) If another reel/unit exists for the file, a reel/unit swap occurs, the current volume pointer is updated to point to the next reel/unit existing in the file, the standard beginning reel/unit label procedure is executed, and the file position indicator is set to one less than the number of the first record existing on the new current volume. If no data records exist for the current volume, another reel/unit swap occurs.

Output Files (Reel/Unit Media):

The following operations take place:

1) The standard ending reel/unit label procedure is executed.

2) A reel/unit swap. The current volume pointer is updated to point to the new reel/unit.

3) The standard beginning reel/unit label procedure is executed.

4) The next executed WRITE statement that references that file directs the next logical data record to the next reel/unit of the file.

Input Files, Input-Output Files, and Output Files (Non-Reel/Unit Media):

Execution of this statement is considered successful. The file remains in the open mode, the file position indicator is unchanged, and no action takes place except as specified in general rule 4.

Page No.	Description of Change

VII-41 Paragraph 4.3.4 of the OPEN statement in the Sequential I-O module, general rule 8, change to read:

(8) If during the execution of an OPEN statement a file attribute conflict condition occurs, the execution of the OPEN statement is unsuccessful. The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The validation of fixed file attributes may vary depending on the organization and/or storage medium of the file. (See page II-1, File Attributes.)

VII-43 Paragraph 4.3.4 of the OPEN statement in the Sequential I-O module, add the following as the new general rule 26:

(26) If the execution of the OPEN statement is unsuccessful, the physical file is not affected and the following actions take place in the following order:

a. A value is placed in the I-O status associated with file-name to indicate the condition that caused the OPEN statement to be unsuccessful.

b. Any applicable USE AFTER EXCEPTION procedure is executed. (See page VII-50, The USE Statement.)

VII-46 Paragraph 4.4.4 of the READ statement in the Sequential I-O module, replace general rule 10b with the following:

b. If the AT END phrase is specified in the statement causing the condition, control is transferred to the AT END imperative-statement-1. Any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1 is not executed. Execution then continues according to the rules for each statement specified in imperative-statement-1. If a procedure branching or conditional statement that causes explicit transfer of control is executed, control is transferred in accordance with the rules of that statement; otherwise, upon completion of the execution of imperative-statement-1, control is transferred to the end of the READ statement and the NOT AT END phrase, if specified, is ignored.

VII-46 Paragraph 4.4.4 of the READ statement in the Sequential I-O module, general rule 10c, change the first sentence to read: "If the AT END phrase is not specified, a USE AFTER EXCEPTION procedure must be associated with the file connector referenced by file-name-1, and that procedure is executed."

Page No.	Description of Change
VII-46	Paragraph 4.4.4 of the READ statement in the Sequential I-O module, replace general rule 11b with the following: b. If an exception condition that is not an at end condition exists, control is transferred according to rules of the USE statement following the execution of any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1. (See page VII-50, The USE Statement.)
VII-48 (A-19)	Paragraph 4.5.3 of the REWRITE statement in the Sequential I-O module, syntax rule 1, second sentence, change to read: Record-name-1 and identifier-1 must not reference the same storage area.
VII-50	Paragraph 4.6.3 of the USE statement in the Sequential I-O module, syntax rule 3, change to read: (3) The same file-name must not appear in more than one USE AFTER EXCEPTION statement within the same Procedure Division.
VII-52 (A-19)	Paragraph 4.7.3 of the WRITE statement in the Sequential I-O module, syntax rule 1, second sentence, change to read: Record-name-1 and identifier-1 must not reference the same storage area.
VII-53	Paragraph 4.7.4 of the WRITE statement in the Sequential I-O module, replace general rule 9, including subparagraphs a and b, with the following: (9) If, during the successful execution of a WRITE statement with the NOT END-OF-PAGE phrase, the end-of-page condition does not occur, control is transferred to imperative-statement-2 after execution of the input-output operation.
VII-54	Paragraph 4.7.4 of the WRITE statement in the Sequential I-O module, renumber general rules 10 through 17 as new general rules 11 through 18.

Page No.	Description of Change

VII-54 Paragraph 4.7.4 of the WRITE statement in the Sequential I-O module, add the following as the new general rule 10:

(10) If the execution of the WRITE statement is unsuccessful, the I-O status of the file-name associated with record-name-1 is updated and control is transferred according to the rules of the USE statement following the execution of any USE AFTER EXCEPTION procedure applicable to the file-name associated with record-name-1. (See page VII-50, The USE Statement).

VIII-6 Delete paragraph 1.3.7 entitled *The File Attribute Conflict Condition* that appears on the bottom of page VIII-6 and on the top of page VIII-7.

VIII-24 Paragraph 4.4.4 of the OPEN statement in the Relative I-O module, general rule 8, change to read:

(8) If during the execution of an OPEN statement a file attribute conflict condition occurs, the execution of the OPEN statement is unsuccessful. The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The validation of the fixed file attributes may vary depending on the organization and/or storage medium of the file. (See page II-1, File Attributes.)

VIII-25 Paragraph 4.4.4 of the OPEN statement in the Relative I-O module, add the following as the new general rule 19:

(19) If the execution of the OPEN statement is unsuccessful, the physical file is not affected and the following actions take place in the following order:

a. A value is placed in the I-O status associated with file-name to indicate the condition that caused the OPEN statement to be unsuccessful.

b. Any applicable USE AFTER EXCEPTION procedure is executed. (See page VIII-35, The USE Statement.)

Page No.	Description of Change

VIII-28 Paragraph 4.5.4 of the READ statement in the Relative I-O module, general rule 10b, change to read:

b. If the AT END phrase is specified in the statement causing the condition, control is transferred to the AT END imperative-statement-1. Any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1 is not executed. Execution then continues according to the rules for each statement specified in imperative-statement-1. If a procedure branching or conditional statement that causes explicit transfer of control is executed, control is transferred in accordance with the rules of that statement; otherwise, upon completion of the execution of imperative-statement-1, control is transferred to the end of the READ statement and the NOT AT END phrase, if specified, is ignored.

VIII-28 Paragraph 4.5.4 of the READ statement in the Relative I-O module, general rule 10c, change the first sentence to read: "If the AT END phrase is not specified, a USE AFTER EXCEPTION procedure must be associated with the file connector referenced by file-name-1, and that procedure is executed."

VIII-29 Paragraph 4.5.4 of the READ statement in the Relative I-O module, replace general rule 11b with the following:

b. If an exception condition that is not an at end or an invalid key condition exists, control is transferred according to rules of the USE statement following the execution of any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1. (See page VIII-35, The USE Statement.)

VIII-30 (A-19) Paragraph 4.6.3 of the REWRITE statement in the Relative I-O module, syntax rule 1, second sentence, change to read:

Record-name-1 and identifier-1 must not reference the same storage area.

VIII-35 Paragraph 4.8.3 of the USE statement in the Relative I-O module, syntax rule 3, change to read:

(3) The same file-name must not appear in more than one USE AFTER EXCEPTION statement within the same Procedure Division.

Page No.	Description of Change
VIII-37 (A-19)	Paragraph 4.9.3 of the WRITE statement in the Relative I-O module, syntax rule 1, second sentence, change to read: Record-name-1 and identifier-1 must not reference the same storage area.
VIII-38	Paragraph 4.9.4 of the WRITE statement in the Relative I-O module, replace general rule 9, including subparagraphs a and b, with the following: (9) Transfer of control following the successful or unsuccessful execution of the WRITE operation depends on the presence or absence of the INVALID KEY and NOT INVALID KEY phrases. (See page VIII-5, The Invalid Key Condition.)
IX-4	Paragraph 1.3.4, I-O status in the Indexed I-O module, item 3b for I-O status 22, change to read: b. I-O Status = 22. An attempt is made either: 1) to write a record that would create a duplicate prime record key, or 2) to write or rewrite a record that would create a duplicate alternate record key when the DUPLICATES phrase is not specified for that alternate record key.
IX-7	Delete paragraph 1.3.7 entitled *The File Attribute Conflict Condition.*
IX-26	Paragraph 4.4.4 of the OPEN statement in the Indexed I-O module, general rule 8, change to read: (8) If during the execution of the OPEN statement a file attribute conflict condition occurs, the execution of the OPEN statement is unsuccessful. The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The validation of the fixed file attributes may vary depending on the organization and/or storage medium of the file. (See page II-1, File Attributes.)

Page No.	Description of Change

IX-27 Paragraph 4.4.4 of the OPEN statement in the Indexed I-O module, add the following as the new general rule 19:

(19) If the execution of the OPEN statement is unsuccessful, the physical file is not affected and the following actions take place in the following order:

a. A value is placed in the I-O status associated with file-name to indicate the condition that caused the OPEN statement to be unsuccessful.

b. Any applicable USE AFTER EXCEPTION procedure is executed. (See page IX-39, The USE Statement.)

IX-30 Paragraph 4.5.4 of the READ statement in the Indexed I-O module, general rule 10b, change to read:

b. If the AT END phrase is specified in the statement causing the condition, control is transferred to the AT END imperative-statement-1. Any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1 is not executed. Execution then continues according to the rules for each statement specified in imperative-statement-1. If a procedure branching or conditional statement that causes explicit transfer of control is executed, control is transferred in accordance with the rules of that statement; otherwise, upon completion of the execution of imperative-statement-1 control is transferred to the end of the READ statement and the NOT AT END phrase, if specified, is ignored.

IX-30 Paragraph 4.5.4 of the READ statement in the Indexed I-O module, general rule 10c, replace the first sentence with the following: "If the AT END phrase is not specified, a USE AFTER EXCEPTION procedure must be associated with the file connector referenced by file-name-1, and that procedure is executed."

IX-31 Paragraph 4.5.4 of the READ statement in the Indexed I-O module, replace general rule 11b with the following:

b. If an exception condition that is not an at end or an invalid key condition exists, control is transferred according to rules of the USE statement following the execution of any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-1. (See page IX-39, The USE Statement.)

Page No.	Description of Change
IX-33 (A-19)	Paragraph 4.6.3 of the REWRITE statement in the Indexed I-O module, syntax rule 1, second sentence, change to read: Record-name-1 and identifier-1 must not reference the same storage area.
IX-39	Paragraph 4.8.3 of the USE statement in the Indexed I-O module, syntax rule 3, change to read: (3) The same file-name must not appear in more than one USE AFTER EXCEPTION statement within the same Procedure Division.
IX-41 (A-20)	Paragraph 4.9.3 of the WRITE statement in the Indexed I-O module, syntax rule 1, second sentence, change to read; Record-name-1 and identifier-1 must not reference the same storage area.
IX-42	Paragraph 4.9.4 of the WRITE statement in the Indexed I-O module, general rule 9, including subparagraphs a and b, change to read: (9) Transfer of control following the successful or unsuccessful execution of the WRITE operation depends on the presence or absence of the INVALID KEY and NOT INVALID KEY phrases. (See page IX-6, The Invalid Key Condition.)
X-7	Paragraph 1.3.8.3, Conventions for Index-Names, change the first sentence to read: "If a data item possessing the global attribute includes a table described with an index-name, that index-name also possesses the global attribute."
X-23	Paragraph 4.5.3 of the EXTERNAL clause, syntax rule 3, change the first sentence to read: "The VALUE clause must not be used in any data description entry that includes, or is subordinate to an entry that includes, the EXTERNAL clause."
X-33	Paragraph 5.4.4 of the EXIT PROGRAM statement, add the following as the new general rule 4: (4) An EXIT PROGRAM statement must not be executed while executing within the range of a declarative procedure in which the GLOBAL phrase is specified, except within a program called while executing within the range of that declarative procedure.

Page No.	Description of Change

XI-10 Paragraph 4.1.4 of the MERGE statement, general rule 7b, add the following as the new last sentence for the paragraph:

If the file referenced by file-name-1 is described with variable length records, the size of any record written to file-name-1 is the size of that record when it was read from file-name-2 or file-name-3, regardless of the content of the data item referenced by the DEPENDING ON phrase of either a RECORD IS VARYING clause or an OCCURS clause specified in the sort-merge file description entry for file-name-1.

XI-11 Paragraph 4.1.4 of the MERGE statement, general rule 7c, add the following as a new last sentence after the words "... the last statement in the output procedure.":

For a relative file, the content of the relative key data item is undefined after the execution of the MERGE statement.

XI-11 Paragraph 4.1.4 of the MERGE statement, general rule 7, add the following as a new last paragraph for general rule 7:

The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the file description entry for file-name-2 or file-name-3 is undefined upon completion of the MERGE statement.

XI-11 Paragraph 4.1.4 of the MERGE statement, general rule 11b, add the following as the new last sentence for the first paragraph:

If the file referenced by file-name-4 is described with variable length records, the size of any record written to file-name-4 is the size of that record when it was read from file-name-1, regardless of the content of the data item referenced by the DEPENDING ON phrase of either a RECORD IS VARYING clause or an OCCURS clause specified in the file description entry for file-name-4.

XI-12 Paragraph 4.1.4 of the MERGE statement, general rule 11, last paragraph, sixth line, change in part from "... any USE AFTER STANDARD EXCEPTION/ERROR procedure specified for the file is executed; ..." to "... any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-4 is executed; ...".

Page No.	Description of Change

XI-12 Paragraph 4.1.4 of the MERGE statement, general rule 11, add the following as a new last paragraph for general rule 11:

The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the sort-merge file description entry for file-name-1 is undefined upon completion of the MERGE statement for which the GIVING phrase is specified.

XI-13 (A-20) Paragraph 4.2.3 of the RELEASE statement in the Sort-Merge module, syntax rule 3, second sentence, change to read:

Record-name-1 and identifier-1 must not reference the same storage area.

XI-19 Paragraph 4.4.4 of the SORT statement, beginning of general rule 9b to the end of general rule 9, replace with the following:

b. The logical records are obtained and released to the sort operation. Each record is obtained as if a READ statement with the NEXT and the AT END phrases had been executed. If the file referenced by file-name-1 is described with variable length records, the size of any record released to file-name-1 is the size of that record when it was read from file-name-2, regardless of the content of the data item referenced by the DEPENDING ON phrase of either a RECORD IS VARYING clause or an OCCURS clause specified in the sort-merge file description entry for file-name-1.

c. The processing of the file is terminated. The termination is performed as if a CLOSE statement without optional phrases had been executed. This termination is performed before the file referenced by file-name-1 is sequenced by the SORT statement. For a relative file, the content of the relative key data item associated with file-name-2 is undefined after the execution of the SORT statement if file-name-2 is not referenced in the GIVING phrase.

These implicit functions are performed such that any associated USE AFTER EXCEPTION/ERROR procedures are executed; however, the execution of such a USE procedure must not cause the execution of any statement manipulating the file referenced by, or accessing the record area associated with, file-name-2.

The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the file description entry for file-name-2 is undefined upon completion of the SORT statement.

XI-20 Paragraph 4.4.4 of the SORT statement, general rule 12b, first paragraph, last sentence, change "phrases had been execution." to "phrases had been executed.".

Page No.	Description of Change

XI-20 Paragraph 4.4.4 of the SORT statement, general rule 12b, add the following as the new last sentence in the first paragraph:

If the file referenced by file-name-3 is described with variable length records, the size of any record written to file-name-3 is the size of that record when it was read from file-name-1, regardless of the content of the data item referenced by the DEPENDING ON phrase of either a RECORD IS VARYING clause or an OCCURS clause specified in the file description entry for file-name-3.

XI-20 Paragraph 4.4.4 of the SORT statement, general rule 12, last paragraph, last sentence beginning with "On the first attempt", replace with the following:

On the first attempt to write beyond the externally defined boundaries of the file, any USE AFTER EXCEPTION procedure associated with the file connector referenced by file-name-3 is executed; if control is returned from that USE procedure or if no such USE procedure is specified, the processing of the file is terminated as in paragraph 12c above.

The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the sort-merge file description entry for file-name-1 is undefined upon completion of the SORT statement for which the GIVING phrase is specified.

XII-3 Paragraph 2.4 of the COPY statement, general rule 5b, add the following as the new last sentence:

Each lowercase letter is equivalent to the corresponding uppercase letter as specified for the COBOL character set. (See page III-3, Definition of COBOL Character Set.)

XII-5 Paragraph 2.4 of the COPY statement, add the following as a new general rule 12:

(12) The text produced as a result of the complete processing of a COPY statement must not contain a COPY statement.

XII-7 Paragraph 3.4 of the REPLACE statement, general rule 5, change to read:

(5) The text produced as a result of the processing of a REPLACE statement must contain neither a COPY statement nor a REPLACE statement.

Page No.	Description of Change
XII-7	Paragraph 3.4 of the REPLACE statement, general rule 6b, add the following as the new last sentence: Each lowercase letter is equivalent to the corresponding uppercase letter as specified for the COBOL character set. (See page III-3, Definition of COBOL Character Set.)
XII-7	Paragraph 3.4 of the REPLACE statement, general rule 7, change the last sentence to read: A comment line or blank line in source program text is not placed into the resultant program if that comment line or blank line appears within the sequence of text words that match pseudo-text-1.
XIII-34	Paragraph 3.10.8 entitled *Body Group Presentation Rules*, rule 4b, second paragraph, fourth line, change "current page group" to "current body group".
XIII-71	Paragraph 4.5.4 of the OPEN statement in the Report Writer module, general rule 6, change to read: (6) If during the execution of an OPEN statement a file attribute conflict condition occurs, the execution of the OPEN statement is unsuccessful. The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The validation of fixed file attributes may vary depending on the organization and/or storage medium of the file. (See page II-1, File Attributes.)
XIII-78	Paragraph 4.9.3 of the USE BEFORE REPORTING statement, syntax rule 2, change to read: (2) Identifier-1 must reference a report group. The same identifier-1 must not appear in more than one USE BEFORE REPORTING statement within the same Procedure Division.
(A-27)	Paragraph 1.2.2 entitled *Value Returned by a Function*, replace the last sentence with the following: If, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, then the result of such a reference is undefined.

Page No.	Description of Change

(A-28) Paragraph 2.2 entitled *Arguments*, first paragraph after the numbered item 4, replace the last sentence with the following:

If, at the time a function is referenced, the arguments specified for that reference do not have values within the permissible range, then the result of such a reference is undefined.

(A-29) Paragraph 2.2 entitled *Arguments*, last paragraph, replace the last sentence with the following:

The evaluation of an ALL subscript must result in at least one argument, otherwise the result of the reference to the function-identifier is undefined.

(A-52) Paragraph 2.23.3 of the MAX function in the Intrinsic Function module, argument rule 1, change to read:

(1) If more than one argument-1 is specified, all arguments must be of the same class with the exception that mixing of arguments of alphabetic and alphanumeric classes is allowed.

(A-56) Paragraph 2.27.3 of the MIN function in the Intrinsic Function module, argument rule 1, change to read:

(1) If more than one argument-1 is specified, all arguments must be of the same class with the exception that mixing of arguments of alphabetic and alphanumeric classes is allowed.

(A-61) Paragraph 2.32.3 of the ORD-MAX function in the Intrinsic Function module, argument rule 1, change to read:

(1) If more than one argument-1 is specified, all arguments must be of the same class with the exception that mixing of arguments of alphabetic and alphanumeric classes is allowed.

(A-62) Paragraph 2.33.3 of the ORD-MIN function in the Intrinsic Function module, argument rule 1, change to read:

(1) If more than one argument-1 is specified, all arguments must be of the same class with the exception that mixing of arguments of alphabetic and alphanumeric classes is allowed.

Page No.	Description of Change
XVII-13 (A-22)	Paragraph 3.5, last paragraph, last line, change "X3.23A-1989" to "X3.23a-1989".
XVII-13 (A-22)	Add the following after paragraph 3.5:

3.6 AMERICAN NATIONAL STANDARD COBOL 1985, AMENDMENT 2

The X3J4 COBOL Technical Committee of the Accredited Standards Committee X3 was charged with the responsibility of developing an amendment to ANSI X3.23-1985 as a means of resolving ambiguities and correcting errors found therein. In December 1987, X3J4 began the task of preparing an amendment to ANSI X3.23-1985. Errors were corrected which were found during the interpretation of ANSI X3.23-1985.

In January 1990 X3J4 approved the content and format for the draft proposed Amendment 2 to ANSI X3.23-1985 and recommended to X3 that it be published for public review and comment. X3J4 held this public review and comment period during which comments were received from the data processing community on the content of the draft proposed Amendment 2 to ANSI X3.23-1985. X3J4 reviewed and responded to all comments received during this public review period.

Additional errors in X3.23-1985 and X3.23a-1989 (the Intrinsic Function Amendment) were identified and corrections were included in the draft proposed Correction Amendment as a result of public review comments. X3J4 agreed that only ambiguities and errors which were identified by October 17, 1990 would be considered for inclusion in the amendment.

XVII-15 (A-22)	Paragraph 4.4, change the heading to read:

4.4 ISO 1989:1985/Amd.1:1992

Page No.	Description of Change

XVII-15 (A-22) Add the following after paragraph 4.4:

4.5 ISO 1989:1985/Amd.2:1994

In February 1987, the convenor of ISO Technical Committee 97, Subcommittee 22, Working Group 4 on COBOL, forwarded to the SC22 Secretariat the results of a letter ballot to subdivide the COBOL project and produce an amendment to ISO 1989:1985 to resolve identified ambiguities and correct identified errors. At the same time ISO/TC97/SC22 Working Group 4 recommended that the United States be requested to provide draft documents for the amendment. The recommendations of ISO/TC97/SC22 Working Group 4 were approved and the work of developing the amendment was assigned to X3J4.

During X3J4's work on Amendment 2, close and continuous liaison was maintained with the international community through ISO/IEC JTC1/SC22 Working Group 4. The draft document was presented for review and comment to ISO/IEC JTC1/SC22 in March 1990 as a draft proposed Amendment 2 to ISO 1989:1985 for COBOL. In October 1990, ISO/IEC JTC1/SC22 Working Group 4 for COBOL approved a resolution to revise the draft document in response to international comments. Working Group 4 for COBOL agreed that only ambiguities and errors which were identified by October 17, 1990 would be considered for inclusion in the amendment.

XVII-19 (A-22) Change the entry for "Reference-modifier" to "Reference modification".

XVII-42 Change the title of paragraph 2.1 to:

2.1 SUBSTANTIVE CHANGES NOT AFFECTING EXISTING PROGRAMS (THIRD STANDARD COBOL)

XVII-50 Insert the following as the new paragraph 2.1A following item 107 of paragraph 2.1:

2.1A SUBSTANTIVE CHANGES NOT AFFECTING EXISTING PROGRAMS (INTRINSIC FUNCTION AMENDMENT)

The following is a list of the changes of substance introduced by the Intrinsic Function Amendment that are new features not impacting existing programs.

Page No.	Description of Change
XVII-50 (A-23)	Renumber item 108 as the new item 1 under paragraph 2.1A.
XVII-50	Insert the following as the new paragraph 2.1B following item 1 of paragraph 2.1A:

2.1B SUBSTANTIVE CHANGES NOT AFFECTING EXISTING PROGRAMS (CORRECTION AMENDMENT)

The following is a list of the changes of substance introduced by the Correction Amendment that are new features not impacting existing programs.

(1) **INSPECT statement (1 NUC).** For Format 1 of the INSPECT statement, identifier-1 is a sending data item.

(2) **INSPECT statement (2 NUC).** The item in the BEFORE or AFTER phrase of an INSPECT statement with the REPLACING CHARACTERS phrase may be longer than one character.

(3) **MAX function (1 ITR).** If more than one argument is specified, a mixture of alphabetic and alphanumeric arguments is permitted.

(4) **MIN function (1 ITR).** If more than one argument is specified, a mixture of alphabetic and alphanumeric arguments is permitted.

(5) **ORD-MAX function (1 ITR).** If more than one argument is specified, a mixture of alphabetic and alphanumeric arguments is permitted.

(6) **ORD-MIN function (1 ITR).** If more than one argument is specified, a mixture of alphabetic and alphanumeric arguments is permitted.

XVII-51	Change the title of paragraph 2.2 to:

2.2 SUBSTANTIVE CHANGES POTENTIALLY AFFECTING EXISTING PROGRAMS (THIRD STANDARD COBOL)

XVII-71 (A-24)	Substantive change 39, entitled *Reserved words*, delete the word FUNCTION.
XVII-75	Change item 40g concerning I-O status 37 to read:

g. **I-O status = 37.** A permanent error exists because an OPEN statement is attempted on a file and that file will not support the open mode specified in the OPEN statement. (See paragraph 2.2B, item 22, for further explanation.)

Page No.	Description of Change

XVII-80 Paragraph 2.2, insert the following items after item 42d:

(43) **RECORD clause (1 SEQ, 1 REL, 1 INX, 1 RPW).** Files specified using the RECORD CONTAINS integer-1 CHARACTERS clause are always fixed format files that contain only fixed length records. (See paragraph 2.2B, item 3, for further explanation.)

(44) **Floating insertion editing (1 NUC).** If all numeric character positions in the PICTURE character-string are represented by a floating insertion character, at least one of the insertion characters must be to the left of the decimal point. (See paragraph 2.2B, item 4, for further explanation.)

(45) **Size error condition without ON SIZE ERROR phrase (1 NUC).** If a size error condition occurs and an ON SIZE ERROR phrase has not been specified, the program will continue and the value of any affected resultant identifier is undefined. (See paragraph 2.2B, item 8, for further explanation.)

(46) **Multiple results in arithmetic statements (1 NUC).** A temporary data item is always used in evaluating arithmetic statements with multiple receiving fields. (See paragraph 2.2B, item 9, for further explanation.)

(47) **COPY REPLACING statement (2 STM).** The characters right and left parentheses are always text words in third Standard COBOL even when they are not separators. Therefore, they will be processed as individual text words by a COPY REPLACING statement. (See paragraph 2.2B, item 12, for further explanation.)

XVII-80 Insert the following as the new paragraph 2.2A following the last new item at the end of paragraph 2.2:

2.2A SUBSTANTIVE CHANGES POTENTIALLY AFFECTING EXISTING PROGRAMS (INTRINSIC FUNCTION AMENDMENT)

The following is a list of the changes of substance introduced by the Intrinsic Function Amendment that are changes that could impact existing programs.

Page No. **Description of Change**

(1) **Reserved word FUNCTION (1 NUC).** The word FUNCTION must be recognized as a reserved word if the Intrinsic Function module is included in an implementation. (This change was subsequently modified; see paragraph 2.2B, item 1.)

Justification:

The benefits to be derived from the additional facility provided through the addition of the reserved word FUNCTION were considered to outweigh the inconvenience caused by removing this word from the set of user-defined words.

The X3J4 COBOL Technical Committee believes that any one-time difficulties caused by this change ultimately will be more than offset by the facility provided by this change.

XVII-80 Insert the following as the new paragraph 2.2B after the last item of the new paragraph 2.2A:

2.2B SUBSTANTIVE CHANGES POTENTIALLY AFFECTING EXISTING PROGRAMS (CORRECTION AMENDMENT)

The following is a list of the changes of substance introduced by or identified in the Correction Amendment that are changes that could impact existing programs.

(1) Reserved word FUNCTION (1 NUC). The module-dependent reserved word FUNCTION in the Intrinsic Function Amendment was changed to a reserved word in the Correction Amendment.

Justification:

The word FUNCTION was specified as a module-dependent reserved word in the Intrinsic Function Amendment. The Correction Amendment eliminates the classification of module-dependent reserved words and identifies the word FUNCTION as a reserved word.

The X3J4 COBOL Technical Committee believes that any one-time difficulties caused by this change ultimately will be more than offset by the improved portability possible with this change.

(2) **Figurative constant (1 NUC).** The length of a figurative constant associated with a zero-length group item is defined to be one.

Page No. **Description of Change**

Justification:

In third Standard COBOL, the length of a figurative constant is zero when associated with a zero-length group item. This definition incorrectly resulted in equality for a comparison between a zero-length group item and a figurative constant.

Given the following data descriptions in third Standard COBOL:

```
DATA DIVISION.
01  K  PIC 9  VALUE IS 0.
01  A.
    02  B  PIC X  OCCURS 0 TO 3 TIMES
               DEPENDING ON K.
```

The condition A = ZERO is true.

The condition A = 0 is false.

The condition A = ALL "1" is true.

The condition A = "1" is false.

In the Correction Amendment, the length of a figurative constant associated with a zero-length group item is defined to be one. With this new definition, the comparisons of a zero-length group item in the above example all give a result of false, which is the result given for a class test of a zero-length group item. The X3J4 COBOL Technical Committee believes this is what many users would expect.

Although the X3J4 COBOL Technical Committee recognizes that some users may experience some difficulty with this clarification, X3J4 believes that this change ultimately will help promote program portability.

(3) **RECORD clause (1 SEQ, 1 REL, 1 INX, 1 RPW).** Files specified using the RECORD CONTAINS integer-1 CHARACTERS clause are always fixed format files that contain only fixed length records.

This substantive change was introduced by third Standard COBOL; however, it was not identified as a substantive change until amendment 2. Therefore, it has been listed here and as item 43 in paragraph 2.2, Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Page No. **Description of Change**

Justification:

In second Standard COBOL, it is clear that all records must have the same length when associated with a file specified with the RECORD CONTAINS integer-1 CHARACTERS clause. But it is unclear whether or not the associated file is treated as a fixed format file. Third Standard COBOL clarifies the situation by specifying that: (1) integer-1 must specify the number of character positions contained in each record, and (2) the file must include fixed length records (that is, a fixed format file).

This clarification might impact implementations that support variable format files containing only records of the same length. Although files using this format of the RECORD clause could have either variable or fixed format in second Standard COBOL, files using this format of the RECORD clause must have fixed format in third Standard COBOL.

Some implementations already support files using this format of the RECORD clause as fixed format files. Thus, this change will only impact source programs that use this format of the RECORD clause in the following situations:

a. where an implementation supported a variable format file with records all of the same length,

b. where files specified with this format of the RECORD clause were implemented as variable format files, and

c. where the same file needs to be accessed by a second Standard COBOL program using this format of the RECORD clause and by a third Standard COBOL program using this format of the RECORD clause (or in cases where a third Standard COBOL program needs to interface with a second Standard COBOL external implementor file definition or interface).

The X3J4 COBOL Technical Committee believes that the improved portability derived from this clarification will warrant the compatibility problems that it might cause.

(4) **Floating insertion editing (1 NUC).** If all numeric character positions in the PICTURE character-string are represented by a floating insertion character, at least one of the insertion characters must be to the left of the decimal point.

This substantive change was introduced by third Standard COBOL; however, it was not identified as a substantive change until amendment 2. Therefore, it has been listed here and as item 44 in paragraph 2.2, Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Page No. **Description of Change**

Justification:

In second Standard COBOL, a PICTURE character-string could be represented with all floating insertion characters without any character positions to the left of the actual or implied decimal point. However, when the data item had either a value of zero or significant digits to the right of the decimal point, the output would not use the editing character.

The following code was valid in second Standard COBOL but probably did not produce the results that the programmer expected.

```
01  FIELD-A  PIC V+(5).
01  FIELD-B  PIC .$$.

PROCEDURE DIVISION.
    MOVE ZERO TO FIELD-A
    MOVE .123 TO FIELD-B
```

This syntax, although valid, did not produce the expected results; therefore, this syntax has been explicitly prohibited in third Standard COBOL. Programs with this syntax will need to be changed.

The X3J4 COBOL Technical Committee believes that few conforming programs will be affected by this change in third Standard COBOL as most implementations have already included this change or an extension which supports the previous syntax.

(5) **Conventions for index-names (2 IPC).** The reference to the external attribute applying to an index has been removed.

Justification:

Third Standard COBOL specifies that an index associated with a table that possesses the external attribute also possesses the external attribute. However third Standard COBOL does not clarify how the correspondence of an external index is to be established at execution time. For tables defined in the Linkage Section, third Standard COBOL specifically states that there is no correspondence between an index in the called and calling programs.

Page No. **Description of Change**

In the Correction Amendment, the reference to the external attribute of an index has been removed. The X3J4 COBOL Technical Committee believes that the purposes of standardization are better served by not making an index external rather than by attempting to define the correspondence rules. This approach makes consistent the treatment of an index associated with an external data record and an index associated with parameters passed to a called program. Since most implementors had difficulty with the lack of correspondence rules, the X3J4 COBOL Technical Committee believes that few existing programs depend on the existence of an external index.

(6) **Comparison involving index-names (1 NUC).** Comparison of an index-name to either a nonnumeric data item or a nonnumeric literal is not permitted.

Justification:

In third Standard COBOL, the comparison rules involving an index-name and either a data item or a literal could be interpreted to permit comparison of an index-name to either a nonnumeric data item or a nonnumeric literal.

In the Correction Amendment, the comparison rules do not permit comparison of an index-name to either a nonnumeric data item or a nonnumeric literal. Programs that rely upon comparing an index-name to either a nonnumeric data item or a nonnumeric literal will be affected by this change in the Correction Amendment.

The X3J4 COBOL Technical Committee believes that few implementations actually permit this type of comparison; therefore few existing programs will be affected.

(7) **Class condition (1 NUC).** The result of any class test applied to a zero-length group item is:

a. always true if the word NOT is part of the class condition;

b. otherwise, always false.

Justification:

The result of a class condition applied to a zero-length group item is not defined in third Standard COBOL. A zero-length group item does not contain characters belonging to any class; therefore, the Correction Amendment defines the result of any class condition without the word NOT when applied to a zero-length group item as always false. The result of any class condition with the word NOT when applied to a zero-length group item is always true.

Page No. **Description of Change**

For example, third Standard COBOL does not define what would happen when the following code is executed:

```
01  GROUP-ITEM-1.
    05  ODO-DEPEND  PIC 9.
    05  0D0-TABL.
        10  ELEM  OCCURS 0 TO 9 TIMES
                  DEPENDING ON ODO-DEPEND
                  PIC X.

PROCEDURE DIVISION.
    MOVE ZERO TO ODO-DEPEND
    IF ODO-TABL ALPHABETIC
        DISPLAY "MAYBE"
    ELSE
        DISPLAY "YES"
    END-IF

    IF ODO-TABL NOT ALPHABETIC
        DISPLAY "YES"
    ELSE
        DISPLAY "MAYBE"
    END-IF

    IF ODO-TABL NUMERIC AND ODO-TABL ALPHABETIC
        DISPLAY "MAYBE"
    ELSE
        DISPLAY "YES"
    END-IF

    EVALUATE ODO-TABL NOT NUMERIC
        WHEN NOT ODO-TABL NUMERIC
            DISPLAY  "YES"
        WHEN OTHER
            DISPLAY "MAYBE"
    END-EVALUATE
```

In third Standard COBOL a conforming implementation might have displayed zero, one, two, three, or four occurrences of the word MAYBE. In amended third Standard COBOL, a conforming implementation must display four occurrences of the word YES.

The X3J4 COBOL Technical Committee believes that few programs will be affected by this change in the Correction Amendment. Programs that rely upon another result of such class tests will require changes.

Page No. **Description of Change**

(8) **Size error condition without ON SIZE ERROR phrase (1 NUC).** If a size error condition occurs and an ON SIZE ERROR phrase has not been specified, the program will continue and the value of any affected resultant identifier is undefined.

This substantive change was introduced by third Standard COBOL; however, it was not identified as a substantive change until amendment 2. Therefore, it has been listed here and as item 45 in paragraph 2.2, Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Justification:

In second Standard COBOL, the rules for transfer of control were not explicit when a size error condition occurred and an ON SIZE ERROR phrase was not specified. The rules for this situation have been made explicit in third Standard COBOL to ensure that execution will continue using the standard rules for transfer of control. For example:

```
01  DIGIT1  PIC 9  VALUE 9.
01  DIGIT2  PIC 99  VALUE 9.
01  VALUE-ZERO  PIC 9  VALUE 0.

PROCEDURE DIVISION.
STATEMENT-1.
    ADD +1 TO DIGIT1 DIGIT2.
STATEMENT-2.
    DIVIDE VALUE-ZERO INTO DIGIT1 DIGIT2.
```

In second Standard COBOL, most implementations allowed the program to continue execution after performing the statement in paragraph STATEMENT-1. However, some implementations terminated execution during execution of the statement in paragraph STATEMENT-2. In third Standard COBOL, the rules have been clarified to insure that the program will continue execution after both of these statements. Furthermore, the rules now explicitly state that the value in DIGIT2 will be defined as 10 after executing STATEMENT-1. On the other hand the value of DIGIT1 after the execution of STATEMENT-1 and both DIGIT1 and DIGIT2 after the execution of STATEMENT-2 will be undefined.

Although the X3J4 COBOL Technical Committee recognizes that some users may experience difficulties with this clarification, the X3J4 COBOL Technical Committee believes that this change ultimately will help promote program portability.

Page No. **Description of Change**

(9) **Multiple results in arithmetic statements (1 NUC).** A temporary data item is always used in evaluating arithmetic statements with multiple receiving fields.

This substantive change was introduced by third Standard COBOL; however, it was not identified as a substantive change until amendment 2. Therefore, it has been listed here and as item 46 in paragraph 2.2, Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Justification:

In third Standard COBOL, a temporary data item is always used in evaluating arithmetic statements with multiple receiving fields. Thus the result of the statement

```
MULTIPLY NUM-1 BY NUM-1 NUM-2
```

is equivalent to

```
MOVE NUM-1 TO TEMP
MULTIPLY TEMP BY NUM-1
MULTIPLY TEMP BY NUM-2
```

Thus an initial value of 2 in field NUM-1 and 3 in field NUM-2 would give a result of 6 in field NUM-2 in third Standard COBOL.

In second Standard COBOL, it was not defined whether a temporary data item was used when no arithmetic was necessary to derive an intermediate operand. Thus an implementation of second Standard COBOL could have interpreted the statement

```
MULTIPLY NUM-1 BY NUM-1 NUM-2
```

as being equivalent to

```
MULTIPLY NUM-1 BY NUM-1
MULTIPLY NUM-1 BY NUM-2
```

Thus an initial value of 2 in field NUM-1 and 3 in field NUM-2 would give a result of 12 in field NUM-2 in this example of an implementation of second Standard COBOL.

This change was made in third Standard COBOL to define an undefined situation in a way consistent with similar statements that have clearly defined results. Only programs that use the same item as a sending and receiving operand in an arithmetic statement with multiple receiving operands and that ran on an implementation that treated multiple results as in the second example will be affected.

Page No. **Description of Change**

The X3J4 COBOL Technical Committee believes that most implementors have already adopted this specification and that programs should not be affected by this change.

(10) **CLOSE statement (1 SEQ).** The status of the file position indicator following the execution of a CLOSE statement with a REEL or UNIT phrase is defined.

Justification:

Third Standard COBOL does not define the status of the file position indicator following the execution of a CLOSE statement with a REEL or UNIT phrase.

In the Correction Amendment, the state of the file position indicator following the execution of a CLOSE statement with a REEL or UNIT phrase is defined. If a next reel/unit does exist for the input file or the input-output file, the file position indicator is set to one less than the number of the first record existing on the new current volume. If a next reel/unit does not exist for the input file or the input-output file, the file position indicator indicates that no next reel/unit exists. The X3J4 COBOL Technical Committee believes that this definition of a previously undefined situation is one that most users would expect.

(11) **COPY statement (2 STM).** Lowercase letters are considered to be equivalent to their corresponding uppercase letters when used in text words.

Justification:

Third Standard COBOL does not explicitly state that lowercase letters are considered to be equivalent to their corresponding uppercase letters when used in text words.

The Correction Amendment clarifies that lowercase letters are treated consistently in text words and COBOL words. The X3J4 COBOL Technical Committee believes that most implementors have already adopted this specification and that programs should not be affected by this change.

(12) **COPY REPLACING statement (2 STM).** The characters right and left parentheses are always text words in third Standard COBOL even when they are not separators. Therefore, they will be processed as individual text words by a COPY REPLACING statement.

Page No. **Description of Change**

This substantive change was introduced by third Standard COBOL; however, it was not identified as a substantive change until amendment 2. Therefore, it has been listed here and as item 47 in paragraph 2.2, Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Justification:

Consider an example where the following library text has the text-name COPY-1:

```
01  TABL.
    05  TABL-ENTRIES  OCCURS 3 TIMES.
        10  ELEM  PIC X(01).

PROCEDURE DIVISION.
MAINLINE.
    MOVE ALL "N" TO TABL.
    MOVE ALL "Y" TO ELEM (01).
```

and the source program includes in part:

```
DATA DIVISION.
WORKING-STORAGE SECTION.
    COPY COPY-1
        REPLACING ==(01)==
        BY ==(02)==.
    IF TABL = "NYN"
        DISPLAY "SECOND STANDARD COBOL"
    ELSE IF TABL = "NNYYNN"
        DISPLAY "THIRD STANDARD COBOL".
    STOP RUN.
```

Because (1) second Standard COBOL only includes left and right parentheses as text words when they are also separators, and (2) any punctuation mark in a PICTURE clause is explicitly not a separator, a conforming second Standard COBOL implementation would only replace "(01)" in this example's copied Procedure Division text, not in its copied Data Division text.

However, a conforming third Standard COBOL implementation must process parentheses as text words wherever they occur. Therefore, it would process the three text words "(", "01", and ")" in the copied text and replace them with the pseudo-text "(02)" in both the Data Division and the Procedure Division.

With this change in third Standard COBOL, there is just one defined way to handle parentheses wherever they are used. The X3J4 COBOL Technical Committee believes that few existing conforming programs will be affected by this change.

Page No. **Description of Change**

(13) **EXIT PROGRAM statement (2 IPC).** An EXIT PROGRAM statement must not be executed while executing within the range of a declarative procedure in which the GLOBAL phrase is specified, except within a program called while executing within the range of that declarative procedure.

Justification:

In third Standard COBOL, it is unclear what program or programs would terminate if a global declarative performs a paragraph in a nonglobal declarative that contains an EXIT PROGRAM. The Correction Amendment clarifies this situation specifying that no EXIT PROGRAM may be executed in the range of a global declarative, except within a program called while executing within the range of that global declarative.

This clarification might impact programs that include EXIT PROGRAM in their declaratives and have a global declarative that performs paragraphs or sections in nonglobal declaratives. However, many implementations could give unpredictable results in this situation. The X3J4 COBOL Technical Committee believes that this change will increase program portability and that few if any programs will be impacted.

(14) **IF statement (2 NUC).** IF statements within IF statements are considered matched IF, ELSE, and END-IF ordered combinations, proceeding from left to right. Thus, any ELSE encountered is matched with the nearest preceding IF that either has not already been matched with an ELSE or has not been implicitly or explicitly terminated. Any END-IF encountered is matched with the nearest preceding IF that has not been implicitly or explicitly terminated.

Justification:

Third Standard COBOL does not clearly disallow the matching of an END-IF with an implicitly terminated IF. Any ELSE or END-IF encountered is considered to apply to the immediately preceding IF that has not already been paired with an ELSE or END-IF, but the immediately preceding IF may have already been implicitly terminated by an ELSE phrase of a containing IF statement. In the following example:

Page No. **Description of Change**

```
IF condition-1
    IF condition-2
        IF condition-3  statement-1
        ELSE statement-2
    ELSE statement-3
    END-IF
```

the second ELSE implicitly terminates the third IF statement. The immediately preceding IF for the END-IF is the third IF, but programmers and implementors would not expect the END-IF to apply to the terminated third IF but rather to the unterminated second IF.

The Correction Amendment specifies that an ELSE or an END-IF only matches with a preceding IF that has not been implicitly or explicitly terminated. The Correction Amendment further clarifies that both an ELSE and an END-IF can match with the same IF.

Strict implementation of the third Standard COBOL specification would lead to contradictions of other rules and to nonintuitive nesting of IF statements. Thus, the X3J4 COBOL Technical Committee believes that no existing programs will be affected by this clarification.

(15) **INSPECT statement (1 NUC).** For the purpose of determining its length, identifier-1 in all formats of the INSPECT statement is treated as if it were a sending data item.

Justification:

When a group item contains both a subordinate item described with an OCCURS clause that specifies the DEPENDING ON phrase and the item named in that DEPENDING ON phrase, third Standard COBOL specifies that the evaluation of the length of the group item differs depending on whether the item is a sending or receiving item. When the item is a sending item, the length is evaluated based on the value of the item named in the DEPENDING ON phrase. When the item is a receiving item, the length is the maximum length of the group.

Third Standard COBOL does not specify whether identifier-1 of an INSPECT statement is considered to be a sending item or a receiving item. Thus, in the case described in the previous paragraph, the length of the group item is undefined.

Page No. **Description of Change**

In the Correction Amendment, identifier-1 in all formats of the INSPECT statement is treated as if it were a sending data item for the purpose of determining its length. This causes the length to be consistently evaluated according to the value of the item named in the DEPENDING ON phrase of the OCCURS clause. In the following example, GROUP-ITEM has a length of 10 when referenced in the INSPECT statement.

```
01  GROUP-ITEM.
    02  TABLE-SIZE  PIC S9(5)  VALUE IS 5.
    02  TABLE-ITEM  PIC X  OCCURS 1 TO 20 TIMES
            DEPENDING ON TABLE-SIZE.

PROCEDURE DIVISION.
    INSPECT GROUP-ITEM ...
```

Programs that rely upon other interpretations will be affected by this change in the Correction Amendment. The X3J4 COBOL Technical Committee believes that few existing conforming programs will be affected by this change because most implementations have already implemented this interpretation.

(16) **MERGE statement (1 SRT).** When a merge file or a giving file in the MERGE statement is described with the RECORD IS VARYING clause, each record in the file retains the size it had when read from the using file or the merge file.

Justification:

In third Standard COBOL the size of records in a merge file or a giving file described with the RECORD IS VARYING clause is not defined. The function of the item referenced by the DEPENDING ON phrase is also undefined. The amendment clarifies that the value of the item referenced by the DEPENDING ON phrase is not used in determining the size of records in the file in this situation. It also clarifies that the value of the item is undefined at the end of execution of the statement. The amendment defines the size of each record in a merge file or a giving file described with the RECORD IS VARYING clause to be the size of the record that was read from the using or merge file, respectively. Thus a record retains its original size throughout the merge operation unless it is written to a file with fixed length records.

The amendment improves portability of programs that merge files with variable length records. The X3J4 COBOL Technical Committee believes that the new definition is the one that most users would expect.

Page No. **Description of Change**

(17) **OPEN statement (1 SEQ, 1 REL, 1 INX, 1 RPW).** The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The validation of fixed file attributes may vary depending on the organization and/or storage medium of the file.

Justification:

In third Standard COBOL, detection of a fixed file attribute conflict was mandatory when a file was opened. However, some input-output systems do not have the capability to detect all of the fixed file attribute conflicts. Also it is not always possible to detect conflicts in files written by COBOL processors conforming to second Standard COBOL. Therefore, when programs are moved from implementation to implementation, they may behave differently in situations where potential fixed file attribute conflicts arise.

In the Correction Amendment the implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. The X3J4 COBOL Technical Committee believes that few existing programs will be affected by explicitly recognizing that the implementor defines which of the fixed file attributes are validated during execution of the OPEN statement.

(18) **REPLACE statement (2 STM).** Lowercase letters are considered to be equivalent to their corresponding uppercase letters when used in text words.

Justification:

Third Standard COBOL does not explicitly state that lowercase letters are considered to be equivalent to their corresponding uppercase letters when used in text words.

The Correction Amendment clarifies that lowercase letters are treated consistently in text words and COBOL words. The X3J4 COBOL Technical Committee believes that most implementors have already adopted this specification and that programs should not be affected by this change.

(19) **SORT statement (1 SRT).** When a sort file or a giving file in the SORT statement is described with the RECORD IS VARYING clause, each record in the file retains the size it had when read from the using file or the sort file.

Page No. **Description of Change**

Justification:

In third Standard COBOL the size of records in a sort file or a giving file described with the RECORD IS VARYING clause is not defined. The function of the item referenced by the DEPENDING ON phrase is also undefined. The amendment clarifies that the value of the item referenced by the DEPENDING ON phrase is not used in determining the size of records in the file in this situation. It also clarifies that the value of the item is undefined at the end of execution of the statement. The amendment defines the size of each record in a sort file or a giving file described with the RECORD IS VARYING clause to be the size of the record that was read from the using or sort file, respectively. Thus a record retains its original size throughout the sort operation unless it is written to a file with fixed length records.

The Correction Amendment improves portability of programs which sort files with variable length records. The X3J4 COBOL Technical Committee believes that the new definition is the one that most users would expect.

(20) **WRITE statement (2 SEQ).** Imperative-statement-2 in the NOT END-OF-PAGE phrase is executed only upon successful execution of a WRITE statement in which the end-of-page condition does not occur.

Justification:

Third Standard COBOL specified that imperative-statement-2 in the NOT END-OF-PAGE phrase would be executed when the execution of the WRITE statement is unsuccessful. This specification was inconsistent with the specifications for similar phrases such as the NOT AT END and NOT INVALID KEY phrases in other input-output statements. Furthermore the rule contradicted the rules of the USE statement that stated, for a noncritical error, control is transferred to the next executable statement following the input-output statement whose execution caused the exception. Also contradicted were the rules of the USE statement that stated, for a critical error, the implementor determines what action is taken.

The Correction Amendment removes the contradiction by making the specifications for the NOT END-OF-PAGE phrase in the WRITE statement consistent with rules for similar phrases in other input-output statements.

Existing programs that depend on the execution of imperative-statement-2 for an unsuccessful WRITE statement would require changes. The X3J4 COBOL Technical Committee believes there are few such programs because of the contradiction with the USE statement rules and the inconsistency of the WRITE statement with all the other input-output statements.

Page No. **Description of Change**

(21) **WRITE statement (1 REL, 1 INX).** Imperative-statement-2 in the NOT INVALID KEY phrase is executed only upon successful execution of a WRITE statement in which the invalid key condition does not occur.

Justification:

Third Standard COBOL specified that imperative-statement-2 in the NOT INVALID KEY phrase would be executed when the execution of the WRITE statement is unsuccessful for a reason other than the invalid key condition. This specification was inconsistent with the specifications for the NOT INVALID KEY phrase in other input-output statements. Furthermore the rule contradicted the rules of the USE statement that stated, for a noncritical error, control is transferred to the next executable statement following the input-output statement whose execution caused the exception. Also contradicted were the rules of the USE statement that stated, for a critical error, the implementor determines what action is taken.

The Correction Amendment removes the contradiction by making the specifications for the NOT INVALID KEY phrase in the WRITE statement consistent with rules for the same phrase in other input-output statements.

Existing programs that depend on the execution of imperative-statement-2 for an unsuccessful WRITE statement would require changes. The X3J4 COBOL Technical Committee believes there are few such programs because of the contradiction with the USE statement rules and the inconsistency of the WRITE statement with all the other input-output statements.

(22) **I-O status = 37 (1 SEQ, 1 REL, 1 INX, 1 RPW).** A permanent error exists because an OPEN statement is attempted on a file and that file will not support the open mode specified in the OPEN statement.

This substantive change was introduced by third Standard COBOL; however, the substantive change description was not correct and has been revised in the Correction Amendment. Therefore, it has been listed here and as item 40g in paragraph 2.2 Substantive Changes Potentially Affecting Existing Programs (Third Standard COBOL).

Justification:

This new I-O status value will be returned when a file will not support the mode specified in the OPEN statement; for example, a file assigned to a card reader opened in the output mode. Second Standard COBOL does not specify what happens in this instance. The new I-O status value of 37 allows the user to test for such conditions.

Page No.	Description of Change

(23) **Intrinsic functions (1 ITR).** If, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, or if the evaluation of an ALL subscript does not result in at least one argument, the result of such a reference is undefined.

Justification:

The Intrinsic Function Amendment states that the returned value is undefined if, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, or the evaluation of an ALL subscript does not result in at least one argument. This could be interpreted as requiring an implementation to always return a value and continue processing when a function argument violates specified constraints. This clarification in the Correction Amendment allows an implementation of amended third Standard COBOL to terminate processing in this circumstance. The X3J4 COBOL Technical Committee believes this is what most users would expect. Programs that rely upon continued processing may require change to ensure the correctness of function arguments.

Although the X3J4 COBOL Technical Committee recognizes that some users may experience some difficulty with this clarification, X3J4 believes that the improved reliability of data derived from this clarification warrants the compatibility problems that might arise for some users.

XVII-93 (A-24) Insert items 85 through 87 after item 84 in the Implementor-Defined Language Element List.

(85) **ALPHABET clause in SPECIAL-NAMES paragraph.** The implementor defines the ordinal number within the character code set being specified for each character within the native character set that is not specified by the literal-1 phrase. (See 4.5.4, general rule 4d, subitem 4 on page VI-16.)

(86) **OPEN statement.** The implementor defines which of the fixed file attributes are validated during the execution of the OPEN statement. (See 4.3.4 general rule 8, on page VII-41; 4.4.4, general rule 8, on page VIII-24; 4.4.4, general rule 8, on page IX-26; 4.5.4, general rule 6, on page XIII-71.)

(87) **RANDOM function.** The implementor will specify the subset of the domain of argument-1 values that will yield distinct sequences of pseudo-random numbers. (See 2.35.4, rule 3, on page A-64.)

Page No.	Description of Change
XVII-98 (A-24)	In the change to page XVII-98 inserting item 29 in the Undefined Language Elements List, replace the first sentence after the header *Intrinsic functions* with the following: If, at the time a function is referenced, the arguments specified for that reference do not have values that comply with the specified constraints, then the result of such a reference is undefined.
XVII-98 (A-24)	Insert the following after item 29 in the Undefined Language Element List: (30) **MERGE statement.** For a relative file, the content of the relative key data item is undefined after the execution of the MERGE statement. (See 4.1.4, general rule 17c, on page XI-11.) (31) **MERGE statement.** The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the file description entry for file-name-2 or file-name-3 is undefined upon completion of the MERGE statement. (See 4.1.4, general rule 7, on page XI-11.) (32) **MERGE statement.** The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the sort-merge file description entry for file-name-1 is undefined upon completion of the MERGE statement for which the GIVING phrase is specified. (See 4.1.4, general rule 11, on page XI-12.) (33) **SORT statement.** For a relative file, the content of the relative key data item associated with file-name-2 is undefined after the execution of the SORT statement if file-name-2 is not referenced in the GIVING phrase. (See 4.4.4, general rule 9c, on page XI-19.) (34) **SORT statement.** The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the file description entry for file-name-2 is undefined upon completion of the SORT statement. (See 4.4.4, general rule 9, on page XI-19.) (35) **SORT statement.** The value of the data item referenced by the DEPENDING ON phrase of a RECORD IS VARYING clause specified in the sort-merge file description entry for file-name-1 is undefined upon completion of the SORT statement for which the GIVING phrase is specified. (See 4.4.4, general rule 12, on page XI-20.) (36) **Intrinsic functions.** The evaluation of an ALL subscript must result in at least one argument, otherwise the result of the reference to the function-identifier is undefined. (See 2.2 on page A-29.)

Page No.	Description of Change
XVIII-1	Replace the index on pages XVIII-1 through XVIII-13 with the index on pages B-49 through B-64.

INDEX

'A' PICTURE symbol, VI-31
Abbreviated combined relation conditions, VI-60
ACCEPT MESSAGE COUNT statement, XIV-17
  USE FOR DEBUGGING statement, XV-7
ACCEPT statement, VI-71
  Imperative statement, IV-39
  Mnemonic-name, VI-13
  SPECIAL-NAMES paragraph, VI-13
ACCESS MODE clause
  DYNAMIC, VIII-8, VIII-10, IX-8, IX-10
  RANDOM, VIII-8, VIII-10, IX-8, IX-10
  SEQUENTIAL, VII-7, VII-9, VIII-8, VIII-10, IX-8, IX-10, XIII-3
Access modes, VII-1, VIII-2, IX-2
Accessing data and files, II-19
ACOS function. A-33
ADD CORRESPONDING (ADD CORR) statement, VI-68, VI-73
ADD statement, VI-73
  Composite of operands, VI-69
  COMPUTE statement, VI-76
  Conditional statement, IV-37
  CORRESPONDING (CORR), VI-68
  Data conversion, VI-69
  Decimal alignment, VI-69
  Imperative statement, IV-39
  Maximum operand size, VI-69
  Multiple results, VI-69
Additional language elements, I-7
ADVANCING phrase
  SEND statement, XIV-26
  WRITE statement, VII-52
AFTER phrase
  INSPECT statement, VI-94
  PERFORM statement, VI-109, VI-110
  SEND statement, XIV-26
  WRITE statement, VII-52
Algebraic sign, IV-16
Alignment of data, IV-16
  Synchronization, IV-17
ALL
  Figurative constant, IV-11
  INSPECT statement, VI-94
  SEARCH statement, VI-122
  UNSTRING, VI-136
  USE FOR DEBUGGING statement, XV-5
ALL literal, IV-11
  INSPECT statement, VI-95
  STOP statement, VI-130
  STRING statement, VI-131
  UNSTRING statement, VI-136
ALL PROCEDURES phrase, XV-5
ALL REFERENCES OF phrase, XV-5
ALL subscript, A-9, A-11, A-28
ALPHABET clause, VI-13
Alphabet-name, III-1, IV-6, VI-13
  CODE-SET clause, VII-24
  MERGE statement, XI-8
  SORT statement, XI-16
ALPHABETIC
  Class condition, VI-57
  INITIALIZE statement, VI-92
Alphabetic category, IV-15, VI-29, VI-48, VI-104
Alphabetic class, IV-15, VI-56
Alphabetic data item, VI-29
ALPHABETIC-LOWER, VI-57
ALPHABETIC-UPPER, VI-57
ALPHANUMERIC, VI-92
Alphanumeric category, IV-15, VI-29, VI-48, VI-104
Alphanumeric character, III-1
Alphanumeric class, IV-15
Alphanumeric data item, VI-30
ALPHANUMERIC-EDITED, VI-92
Alphanumeric edited category, IV-15, VI-29, VI-48, VI-104
Alphanumeric edited data item, VI-30
Alphanumeric function, A-8, A-10, A-29
ALSO phrase
  ALPHABET clause, VI-13
  EVALUATE statement, VI-84
ALTER statement, VI-75
  GO TO statement, VI-89
  Imperative statement, IV-39
  Initial state of program, X-10
  Segmentation, XVI-8
  Transfer of control, IV-26
  USE FOR DEBUGGING statement, XV-6
ALTERNATE RECORD KEY clause, IX-8, IX-11

AND
  Abbreviated combined relation condition, VI-61
  Combined condition, VI-59
  Evaluation order, VI-61
  Logical operator, VI-59
  SEARCH statement, VI-122
ANNUITY function, A-34
ANY, VI-84
Arccosine function, A-33
Arcsine function, A-35
Arctangent function, A-36
Area A, IV-41
Area B, IV-41
Argument, A-8, A-11, A-28
Arithmetic expression, VI-51
  COMPUTE statement, VI-76
  EVALUATE statement, VI-84
  Relation condition, VI-54
  SEARCH statement, VI-122
  Sign condition, VI-58
Arithmetic operator, IV-9, VI-52
Arithmetic statement, III-2, VI-69
ASCENDING KEY phrase
  MERGE statement, XI-8
  OCCURS clause, VI-26
  SORT statement, XI-16
ASIN function, A-35
ASSIGN clause
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-7
  Sort-Merge module, XI-2
Asterisk (*) comment line, IV-42
Asterisk (*) PICTURE symbol, VI-32, VI-35
At end condition, VII-5, VIII-6, IX-7
  READ statement, VII-46, VIII-28, IX-30
  RETURN statement, XI-14
AT END phrase
  READ statement, VII-44, VIII-26, IX-28
  RETURN statement, XI-14
  SEARCH statement, VI-122
AT END-OF-PAGE phrase, VII-52
ATAN function, A-36
AUTHOR paragraph, VI-6

'B' PICTURE symbol, VI-31, VI-33
BEFORE phrase
  INSPECT statement, VI-94
  PERFORM statement, VI-109, VI-110
  SEND statement, XIV-26
  WRITE statement, VII-52
BINARY, VI-46
Binary arithmetic operators, VI-52
Blank line, IV-42
BLANK WHEN ZERO clause, VI-22
  PICTURE clause, VI-29
  Report group description entry, XIII-21
  USAGE IS INDEX clause, VI-46
  VALUE clause, VI-49
BLOCK CONTAINS clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-23
Body group presentation rules, XIII-32
Braces, IV-2
Brackets, IV-2
BY
  COPY statement, XII-2
  DIVIDE statement, VI-80
  INITIALIZE statement, VI-92
  INSPECT statement, VI-94
  MULTIPLY statement, VI-107
  PERFORM statement, VI-110
BY CONTENT phrase, X-27
BY REFERENCE phrase, X-27

CALL statement, X-27
  CANCEL statement, X-31
  Conditional statement, IV-37
  EXIT PROGRAM statement, X-33
  Imperative statement, IV-39
  Linkage Section, X-13
  PERFORM statement, VI-120
  Procedure Division header, X-25
  Transfer of Control, II-23, IV-26
Called program, III-2, X-27
Calling program, III-2, X-27
CANCEL statement, X-31
  CALL statement, X-29
  EXIT PROGRAM statement, X-33
  Imperative statement, IV-39
Category of data, IV-15
  Editing, VI-33
  MOVE statement, VI-104
  Nonnumeric literal, IV-10
  Numeric literal, IV-10
  PICTURE clause, VI-29
  VALUE clause, VI-48
CD entry, XIV-3
CD level indicator, III-13, XIV-3, XIV-4
  Reference format, IV-43
Cd-name, III-3, IV-6, XIV-3, XIV-4, XV-5
CF, XIII-55
CH, XIII-55
CHAR function, A-37
Character, IV-4
  Alphabetic, III-1
  Alphanumeric, III-1
  Editing, III-8
  Numeric, III-15
  Punctuation, III-18, IV-4
  Relation, III-19
  Special, III-23

Character function, A-37
Character representation, IV-16
Character set, IV-4
  Restriction, VI-1
Character-string, IV-5
Character substitution, I-8, IV-4
CHARACTERS
  BLOCK CONTAINS clause, VII-23
  INSPECT statement, VI-94
  MEMORY SIZE clause, VI-11
  RECORD clause, VII-30
Choice indicators, IV-2
CLASS clause, VI-13, VI-16, VI-56
Class condition, VI-56, B-9
Class-name, III-3, IV-6, VI-13, VI-16, VI-57
Class of data, IV-15
Clause, III-3
CLOCK-UNITS phrase, VII-17, VII-18
CLOSE statement
  Imperative statement, IV-39
  Indexed I-O module, IX-19
  Relative I-O module, VIII-17
  Report Writer module, XIII-63
  Sequential I-O module, VII-35
  STOP statement, VI-130
COBOL character set, III-3, IV-4
COBOL development, XVII-1
COBOL Journal of Development, XVII-2
COBOL library, XII-2
COBOL reserved words, IV-45
COBOL source program, IV-29, VI-3
COBOL standardization, XVII-11
COBOL word, IV-5, B-5
CODASYL, XVII-1
CODASYL COBOL Committee, XVII-2
CODE clause, XIII-11, XIII-14
CODE-SET clause
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-24
COLLATING SEQUENCE phrase
  ALPHABET clause, VI-15
  MERGE statement, XI-8
  SORT statement, XI-16
Colon, IV-5
  Restriction, VI-1
COLUMN NUMBER clause, XIII-21, XIII-42
Combined condition, VI-59
Comma
  DECIMAL-POINT IS COMMA clause, VI-13, VI-17
  Interchangeable with semicolon, IV-2
  PICTURE symbol, VI-32, VI-33
  Separator, IV-4
Comment-entry, IV-12, VI-6
  DATE-COMPILED paragraph, VI-8
Comment line, IV-42
  Library text, XII-4, XII-7
COMMON clause, X-12
Common program, II-22, X-3, X-12
Communication concepts, II-28
Communication description entry, III-4, IV-34, XIV-2, XIV-3
Communication module, XIV-1
  Element summary, I-36
Communication Section, IV-33, XIV-2
Communication status key conditions, XIV-14
Compile time switch, VI-141, XV-2
Compiler directing sentence, IV-38
Compiler directing statement, IV-38
Complex condition, VI-59
Composite language skeleton, V-1
COMPUTATIONAL (COMP), VI-46
COMPUTE statement, VI-76
  Composite of operands, VI-69
  Conditional statement, IV-37
  Data conversion, VI-69
  Decimal alignment, VI-69
  Imperative statement, IV-39
  Maximum operand size, VI-69
  Multiple results, VI-69
Computer-name, IV-8, VI-10, VI-11
Concepts, II-1
Condition, VI-54
  Abbreviated combined relation condition, VI-60
  Class condition, VI-56
  Combined condition, VI-59
  Complex condition, VI-59
  Condition-name condition, VI-58
  EVALUATE statement, VI-84
  Evaluation rules, VI-61
  IF statement, VI-90
  Negated condition, VI-59
  PERFORM UNTIL statement, VI-109, VI-110
  Relation condition, VI-54
  SEARCH statement, VI-122
  Sign condition, VI-58
  Simple condition, VI-54
  Switch-status condition, VI-58
Condition-name, IV-6, IV-7, IV-24
  Condition-name condition, VI-58
  Conventions, X-6
  Level-number 88, IV-14
  Qualified, IV-19
  RERUN clause, VII-17, VII-18
  SEARCH statement, VI-122
  SET statement, VI-127
  SPECIAL-NAMES paragraph, VI-13
  Subscripted, IV-21
  Switch-status condition, VI-58
  Uniqueness, IV-24
  VALUE clause, VI-49
Condition-name condition, VI-58
Condition-name data description entry, VI-21, VI-25, VI-49
Conditional expression, VI-54
  EVALUATE statement, VI-85
  PERFORM statement, VI-111
  SEARCH statement, VI-123
Conditional phrase, IV-37

Conditional sentence, IV-38
Conditional statement, IV-37
Conditional variable, III-6, VI-58, VI-127
  FILLER, VI-23
Configuration Section, IV-31
  Nucleus, VI-9
Conforming implementation, I-6
Conforming source program, I-9
Contained programs, X-1, X-8
Continuation line, IV-42
Continuation of lines, IV-42
  Comment-entries, VI-6, VI-8
CONTINUE statement, VI-77
Continued line, IV-42
CONTROL clause, XIII-11, XIII-15
Control break
  CONTROL clause, XIII-15
  GENERATE statement, XIII-66
  GROUP INDICATE clause, XIII-44
  TYPE clause, XIII-56
CONTROL FOOTING (CF), XIII-55
CONTROL HEADING (CH), XIII-55
CONVERTING, VI-95
COPY statement, XII-2
  Compiler directing statement, IV-38
CORRESPONDING (CORR) phrase, VI-68
  ADD statement, VI-73
  MOVE statement, VI-103
  ON SIZE ERROR phrase, VI-68
  SUBTRACT statement, VI-134
COS function, A-38
Cosine function, A-38
COUNT IN phrase, VI-136
CR PICTURE symbol, VI-32, VI-34
Currency PICTURE symbol, VI-32, VI-34
Currency sign, III-6, VI-17, VI-34
CURRENCY SIGN clause, VI-13, VI-17, VI-34
Currency symbol, III-6, VI-17, VI-33
CURRENT-DATE function, A-39
Current volume pointer, II-5, VII-2

Data description entry, III-7, VI-20
  Inter-Program Communication module, X-19
  Working-Storage Section, VI-18
Data Division, IV-33
  Communication module, XIV-2
  Element summary, I-48
  Indexed I-O module, IX-16
  Inter-Program Communication module, X-13
  Nucleus, VI-18
  Reference format, IV-43
  Relative I-O module, VIII-14
  Report Writer module, XIII-6
  Sequential I-O module, VII-21
  Sort-Merge module, XI-6
Data Division entries, IV-43
Data-name, III-7, IV-6, VI-23, XIII-20, XIII-43
  Conventions, X-6
  Identifier, IV-23
  Qualified, IV-19
  Subscripted, IV-21
DATA BY phrase, VI-92
Data Item, A-8
DATA RECORDS clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Sequential I-O module, VII-22, VII-25
  Sort-Merge module, XI-7
DATE, VI-72
DATE-COMPILED paragraph, VI-8
DATE-OF-INTEGER function, A-41
DATE-WRITTEN paragraph, VI-6
DAY, VI-72
DAY-OF-INTEGER function, A-42
DAY-OF-WEEK, VI-72
DB PICTURE symbol, VI-32, VI-34
DE, XIII-55
DEBUG-CONTENTS, XV-8
DEBUG-ITEM, IV-9, XV-1, XV-8
DEBUG-LINE, XV-8
Debug module, XV-1
  Element summary, I-38
DEBUG-NAME, XV-8
DEBUG-SUB-1, XV-8
DEBUG-SUB-2, XV-8
DEBUG-SUB-3, XV-8
Debugging line, VI-141
  Library text, XII-4, XII-7
DEBUGGING MODE clause, VI-10, XV-2, XV-3
Debugging section, XV-5
Decimal point
  Actual, VI-32
  Alignment, IV-16
  Assumed, VI-32
DECIMAL POINT IS COMMA clause, VI-13, VI-17, VI-33
Declarative sentence, III-7
Declaratives, IV-35
  Reference format, IV-44
  USE BEFORE REPORTING statement, XIII-78
  USE FOR DEBUGGING statement, XV-5
  USE statement, VII-50, VIII-35, IX-39
Declarative procedures
  Debug module, XV-4
  Indexed I-O module, IX-18
  PERFORM statement, VI-121
  Relative I-O module, VIII-16
  Report Writer module, XIII-62
  Sequential I-O module, VII-34
  USE BEFORE REPORTING statement, XIII-78
  USE FOR DEBUGGING statement, XV-5
  USE statement, VII-50, VIII-35, IX-39
De-editing, VI-104
Definitions, III-1

DELETE statement
  Indexed I-O module, IX-21
  Relative I-O module, VIII-19
DELIMITED BY phrase
  STRING statement, VI-131
  UNSTRING statement, VI-136
Delimited scope statement, IV-39
DELIMITER IN phrase, VI-136
Delimiters
  Character-string, IV-5
  Pseudo-text, III-18, IV-5
DEPENDING phrase
  GO TO statement, VI-89
  OCCURS clause, VI-26
  RECORD clause, VII-30
DESCENDING KEY phrase
  MERGE statement, XI-8
  OCCURS clause, VI-26
  SORT statement, XI-16
DESTINATION COUNT clause, XIV-4
DESTINATION TABLE OCCURS clause, XIV-4
DETAIL (DE), XIII-55
Development of COBOL, XVII-1
Differences between second and third Standard
    COBOL, XVII-16
DISABLE statement, XIV-18
DISPLAY in USAGE clause, VI-46
DISPLAY statement, VI-78
  Figurative constant, IV-11
  Imperative statement, IV-39
  Mnemonic-name, VI-13
DIVIDE statement, VI-80
  Composite of operands, VI-69
  COMPUTE statement, VI-76
  Conditional statement, IV-37
  Data conversion, VI-69
  Decimal alignment, VI-69
  Imperative statement, IV-39
  Maximum operand size, VI-69
  Multiple results, VI-69
Division, III-8, IV-30
Division header, III-8, IV-43
Double character substitution, IV-4
DOWN BY, VI-127
DUPLICATES phrase
  ALTERNATE RECORD KEY clause, IX-8, IX-11
  SORT statement, XI-16
Dynamic access, II-4, VIII-2, IX-2

Editing, VI-104
Editing characters, III-8
Editing rules, VI-33
Editing sign, IV-16, VI-32
Editing sign control symbols, VI-32
EGI, XIV-26
Element summary by COBOL division, I-40
Element summary by module, I-10
Elementary item, IV-14
  MOVE statement, VI-104
  Noncontiguous, VI-18
  PICTURE clause, VI-29
Elementary move, VI-104
Ellipsis, IV-2
ELSE clause, VI-90
EMI, XIV-26
ENABLE statement, XIV-20
END-ADD phrase, VI-73
END-CALL phrase, X-27
END-COMPUTE phrase, VI-76
END DECLARATIVES, IV-35, IV-44
END-DELETE phrase, VIII-19, IX-21
END-DIVIDE phrase, VI-80
END-EVALUATE phrase, VI-84
END-IF phrase, VI-90
END KEY clause, XIV-3, XIV-4
END-MULTIPLY phrase, VI-107
End of COBOL source program, IV-29, VI-3
End of group indicator (EGI), XIV-26
End of message indicator (EMI), XIV-26
END-OF-PAGE phrase, VII-52
END OF REEL/UNIT phrase, VII-15, VII-17
End of segment indicator (ESI), XIV-26
END-PERFORM phrase, VI-109, VI-110
END PROGRAM, VI-5
End program header
  Inter-Program Communication module, X-8
    X-11
  Nucleus, VI-5
  Reference format, IV-44
END-READ phrase, VII-44, VIII-26, IX-28
END-RECEIVE phrase, XIV-23
END-RETURN phrase, XI-14
END-REWRITE phrase, VIII-30, IX-33
END-SEARCH phrase, VI-122
END-START phrase, VIII-33, IX-36
END-STRING phrase, VI-131
END-SUBTRACT phrase, VI-134
END-WRITE phrase, VII-52, VIII-37, IX-41
ENTER COBOL, VI-83
ENTER statement, I-9,VI-83
Entry, III-8
Environment Division, IV-31
  Debug module, XV-3
  Element summary, I-45
  Indexed I-O module, IX-8
  Nucleus, VI-9
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Segmentation module, XVI-4
  Sequential I-O module, VII-6
  Sort-Merge module, XI-2
EQUAL TO relation, VI-54, VI-122, VIII-33,
    IX-36
EOP phrase, VII-52
ERROR KEY clause, XIV-4
Error key values, XIV-16

ERROR PROCEDURES, VII-50, VIII-35, IX-39, X-34, XIII-76
ESI, XIV-26
EVALUATE statement, VI-84
  Conditional expression, VI-54
Exception declarative, II-7
Exception handling, II-6
EXCEPTION PROCEDURE, VII-50, VIII-35, IX-39, X-34, XIII-76
Execution, IV-35
EXIT statement, VI-88
  Imperative statement, IV-39
  PERFORM statement, VI-120
EXIT PROGRAM statement, X-33
  CANCEL statement, X-31
  PERFORM statement, VI-120
  Transfer of control, IV-26
Explicit specifications, IV-25
Exponentiation, VI-52
EXTEND phrase
  OPEN statement, VII-39, VIII-21, IX-23, XIII-70
  USE statement, VII-50, VIII-35, IX-39, XIII-76
Extension language elements, I-8
EXTERNAL clause, X-23
  OCCURS clause, VI-27
External objects, X-2
External switch, IV-28, VI-13, VI-15, VI-127
Externally provided functions, I-7

FACTORIAL function, A-43
FALSE, VI-84
FD level indicator, III-13
  Indexed I-O module, IX-16
  Inter-Program Communication module, X-15
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22
Figurative constant, IV-9, IV-10
  DISPLAY statement, VI-78
  INSPECT statement, VI-95
  MOVE statement, VI-104
  Restriction, VI-2
  STOP statement, VI-130
  STRING statement, VI-131
  UNSTRING statement, VI-136
File, II-1
  Attributes, II-1
  Conceptual characteristics, IV-13
  Physical aspects, IV-13
File connector, III-9
  File control entry, VII-8, VIII-8, IX-9
  OPEN statement, VII-40, VIII-22, IX-24
File control entry, III-9, IV-32
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report writer module, XIII-3
  Sequential I-O module, VII-7
  Sort-Merge module, XI-2
FILE-CONTROL paragraph, III-9, IV-31
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-7
  Sort-Merge module, XI-2
File description entry, III-10, IV-34
  Indexed I-O module, IX-16
  Inter-Program Communication module, X-15
  Relative I-O module, VIII-14
  Report Writer module, XIII-6, XIII-7
  Sequential I-O module, VII-21, VII-22
File-name, III-10, IV-6, VII-7, VIII-8, IX-8, XI-7, XIII-3
File-name conventions, X-6
File operations, II-5
File position indicator, II-5, VII-2, VIII-2, IX-2
File processing, II-3
File Section, IV-33
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-6
  Sequential I-O module, VII-21
  Sort-Merge module, XI-6
  VALUE clause, VI-49
FILE STATUS clause
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-7, VII-10
FILLER, VI-23
  CORRESPONDING, VI-68
  SYNCHRONIZED, VI-45
FINAL
  CONTROL clause, XIII-15
  SUM clause, XIII-52
  TYPE clause, XIII-55
FIRST, VI-94
FIRST DETAIL, XIII-17
Fixed insertion editing, VI-34
Fixed length records, II-3, VII-31
Fixed overlayable segments, XVI-1
Fixed permanent segments, XVI-2
Floating insertion editing, VI-34
FOOTING, VII-27, XIII-17
FOR, VI-94
FOR REMOVAL phrase, VII-35, XIII-63
Format punctuation, IV-2
FROM phrase
  ACCEPT statement, VI-71
  PERFORM VARYING statement, VI-110
  RELEASE statement, XI-13
  REWRITE statement, VII-48, VIII-30, IX-33
  SEND statement, XIV-26
  SUBTRACT statement, VI-134
  WRITE statement, VII-52, VIII-37, IX-41
Function, A-8
Function-identifier, A-8, A-11, A-27
Function-name, A-8, A-10, A-27

Function summary, A-30

General format, IV-1
General rules, IV-3
GENERATE statement, XIII-66
  Imperative statement, IV-39
Generic terms, IV-1
GIVING phrase
  ADD statement, VI-73
  DIVIDE statement, VI-80
  MERGE statement, XI-8
  MULTIPLY statement, VI-107
  SORT statement, XI-16
  SUBTRACT statement, VI-134
GLOBAL clause, X-24, X-34, X-35
  OCCURS clause, VI-27
Global names, X-2
Glossary of COBOL terms, III-1
GO TO statement, VI-89
  Imperative statement, IV-39
  Initial state of program, X-10
  PERFORM statement, VI-120
  SEARCH statement, VI-126
GREATER THAN relation, VI-54, VIII-33, IX-36
Group, IV-14
GROUP INDICATE clause, XIII-21, XIII-44

Hardware dependent language element list, XVII-94
Hardware dependent language elements, I-8
HEADING phrase, XIII-17
High subset, I-6
HIGH-VALUE/HIGH-VALUES, IV-11
  SPECIAL-NAMES paragraph, VI-16
History of COBOL, XVII-1, A-21, B-26
Hyphen (-) continuation line, IV-42

Identification Division, IV-30
  Element summary, I-44
  Inter-Program Communication module, X-12
  Nucleus, VI-6
Identifier, IV-23, IV-35, A-13
IF statement, VI-90
  Conditional expression, VI-54
  Conditional statement, IV-37
  Imperative statement, IV-39
  SEARCH statement, VI-123
Imperative sentence, IV-39
Imperative statement, IV-39
Implementation of Standard COBOL, I-6
Implementor-defined language element list, XVII-87, A-24, B-47
Implementor-defined record types, II-3
Implementor-defined specifications, I-7
Implementor-name, IV-8
  ALPHABET clause, VI-13
  ASSIGN clause, VII-7, VIII-8, IX-8, XI-2, XIII-3
  RECORD DELIMITER clause, VII-13
  RERUN clause, VII-17
  SPECIAL-NAMES paragraph, VI-13
  VALUE OF clause, VII-33
Implicit specifications, IV-25
Implied relational operator, VI-61
Implied subject, VI-61
IN, IV-19, IV-20, XII-2
Incompatible data, VI-70
Indentation, IV-44
Independent segments, XVI-2
Index, III-11, IV-21, IV-22
INDEX in USAGE clause, VI-46
Index data item, VI-47
  Condition-name, VI-21
  CONTROL clause, XIII-15
  Initial value, VI-19
Index data item (continued)
  INITIALIZE statement, VI-92
  MOVE statement, VI-103
  PICTURE clause, VI-29
  Relation condition, VI-56
  SEARCH statement, VI-123
  SET statement, VI-127
Index-name, III-11, IV-6, IV-21
  Conventions, X-7
  OCCURS clause, VI-26
  PERFORM statement, VI-110
  Relation condition, VI-56
  SEARCH statement, VI-122
  SET statement, VI-127
INDEXED BY phrase, VI-26, VI-27, XIV-4
Indexed file, IX-1
Indexed I-O module, IX-1
  Element summary, I-25
Indexed organization, II-2, IX-1, IX-13
Indicator area, IV-41
INITIAL clause, X-12
INITIAL phrase
  Communication description entry, XIV-3, XIV-4
  INSPECT statement, VI-94
Initial program, II-22, X-3
Initial state of program, X-10
Initial values
  File Section, VII-21, XI-6
  Linkage Section, X-14
  Working-Storage Section, VI-19
INITIALIZE statement, VI-92
INITIATE statement, XIII-69
  Imperative statement, IV-39
In-line PERFORM statement, VI-111

Input-Output Section, IV-31
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-6
  Sort-Merge module, XI-2
Input-output statement, III-12
INPUT phrase
  Communication description entry, XIV-3
  DISABLE statement, XIV-18
  ENABLE statement, XIV-20
  OPEN statement, VII-39, VIII-21, IX-23
  USE statement, VII-50, VIII-35, IX-39
INPUT PROCEDURE phrase, XI-16
INSPECT statement, VI-94
  Imperative statement, IV-39
INSTALLATION paragraph, VI-6
Integer, III-12, A-8, B-4
INTEGER function, A-44
  A-8, A-10, A-29
INTEGER-OF-DATE function, A-45
INTEGER-OF-DAY function, A-46
INTEGER-PART function, A-47
Inter-program communication concepts, II-22
Inter-Program Communication module, X-1
  Element summary, I-28
Intermediate subset, I-6
International Organization for
  Standardization (ISO), XVII-14
  A-22, B-26
International standardization of COBOL, XVII-14
  A-22, B-26
Internal objects, X-2
INTO
  DIVIDE statement, VI-80
  READ statement, VII-44, VIII-26, IX-28
  RECEIVE statement, XIV-23
  RETURN statement, XI-14
  STRING statement, VI-131
  UNSTRING statement, VI-136
Intra-program communication, II-26
Intrinsic function concepts, A-7
Intrinsic function module, A-27
  Element summary, A-5
INVALID KEY condition, VIII-5, IX-6
  DELETE statement, VIII-19, IX-21
  READ statement, VIII-26, IX-28
  REWRITE statement, VIII-30, IX-33
  START statement, VIII-33, IX-36
  WRITE statement, VIII-37, IX-41
I-O-CONTROL paragraph, III-10
  Indexed I-O module, IX-15
  Relative I-O module, VIII-13
  Report Writer module, XIII-5
  Sequential I-O module, VII-15
  Sort-Merge module, XI-3
I-O phrase
  Communication description entry, XIV-4
  OPEN statement, VII-39, VIII-21, IX-23
  USE statement, VII-50, VIII-35, IX-39
I-O status, II-7
  FILE STATUS clause, VII-10
  Indexed I-O module, IX-2
  Relative I-O module, VIII-2
  Sequential I-O module, VII-2
I-O TERMINAL phrase, XIV-18, XIV-20

JUSTIFIED (JUST) clause, VI-24
  Condition-name, VI-21
  Figurative constant, IV-11
  Report group description entry, XIII-21
  Standard alignment, IV-17
  STRING statement, VI-131
  USAGE IS INDEX clause, VI-46
  VALUE clause, VI-49

KEY data-names
  MERGE statement, XI-8
  SORT statement, XI-16
KEY phrase
  DISABLE statement, XIV-18
  ENABLE statement, XIV-20
  MERGE statement, XI-8
  OCCURS clause, VI-26
  READ statement, IX-28
  SEARCH statement, VI-123
  SORT statement, XI-16
  START statement, VIII-33, IX-36
Key word, IV-1, IV-8, A-9

LABEL RECORDS clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-26
Language concepts, IV-4
  Element summary, I-41
  Language-name, III-13, IV-8, VI-83
Language structure, IV-4
LAST DETAIL phrase, XIII-17
LEADING
  INSPECT statement, VI-94
  SIGN clause, VI-42
LEFT, VI-44
LENGTH function, A-48
LESS THAN relation, VI-54, VIII-33, IX-36
Level concept, IV-14
Level indicator, IV-43
Level-number, III-13, IV-6, IV-7, IV-14,
    VI-21, VI-25
  Data description entry, VI-20
  Notation, IV-2
  Reference format, IV-44
  Report group description entry, XIII-20,
    XIII-45
Leveling of module elements, I-1
Library, XII-2
Library-name, III-13, IV-6, XII-2
Library text, III-13, XII-2
LIMIT/LIMITS, XIII-17

LINAGE clause, VII-27, X-15
Linage concepts, II-5
LINAGE-COUNTER, VII-5, VII-28, VII-55
  Qualified, IV-19, IV-20
  Special register IV-9
LINE-COUNTER, XIII-1, XIII-13
  Qualified, IV-19, IV-20
  Special register, IV-9
LINE/LINES
  PAGE clause, XIII-17
  SEND statement, XIV-26
  WRITE statement, VII-52
LINE NUMBER clause, XIII-20, XIII-21, XIII-46
LINES AT BOTTOM phrase, VII-27
LINES AT TOP phrase, VII-27
Linkage Section, IV-33, X-13
  VALUE clause, VI-49
List of elements by COBOL division, I-40
List of elements by module, I-10
Literal, IV-9
  ALPHABET clause, VI-13, VI-15
  CURRENCY SIGN clause, VI-13, VI-17
  STOP statement, VI-130
Local names, X-2
LOCK phrase, VII-35, VIII-17, IX-19, XIII-63
LOG function, A-49
LOG-10 function, A-50
Logarithm base e, A-49
Logarithm base 10, A-50
Logical operator, VI-59
  Precedence, VI-60
Logical record, II-2, IV-13, VII-23
LOW-VALUE/LOW-VALUES, IV-11
  SPECIAL-NAMES paragraph, VI-16
LOWER-CASE function, A-51
Lowercase letters, IV-5, VI-29, VI-56

MAX function, A-52
Maximum function, A-52
MCS, II-28
MEAN function, A-53
MEDIAN function, A-54
MEMORY SIZE clause, VI-11
Merge file, XI-1
MERGE statement, XI-8
  Imperative statement, IV-39
  Segmentation, XVI-8
  Transfer of control, IV-26
Merging, II-6
Message concepts, II-32
Message control system, II-28
  STOP statement, VI-130
MESSAGE COUNT clause, XIV-3
MESSAGE DATE clause, XIV-3, XIV-4
MESSAGE phrase, XIV-23
MESSAGE TIME clause, XIV-3, XIV-4
MIDRANGE function, A-55
MIN function, A-56
Minimum function, A-56
Minimum subset, I-6
Minus (-) PICTURE symbol, VI-32, VI-34, VI-35
Mnemonic-name, IV-6, IV-7
  ACCEPT statement, VI-71
  DISPLAY statement, VI-78
  SEND statement, XIV-26
  SET statement, VI-127
  SPECIAL-NAMES paragraph, VI-13
  WRITE statement, VII-52
MOD function, A-57
Module abbreviations, I-4
Module chart, I-5
Module concept, I-1
MODULES, VI-11
MOVE CORRESPONDING (MOVE CORR) statement, VI-103
MOVE statement, VI-103
  CORRESPONDING (CORR), VI-68
  Imperative statement, IV-39
  Index data item, VI-47
MULTIPLE FILE TAPE clause, VII-15, VII-16, XIII-5
Multiple results in arithmetics, VI-69
MULTIPLY statement, VI-107
  Composite of operands, VI-69
  COMPUTE statement, VI-76
  Conditional statement, IV-37
MULTIPLY statement (continued)
  Data conversion, VI-69
  Decimal alignment, VI-69
  Imperative statement, IV-39
  Maximum operand size, VI-69
  Multiple results, VI-69

NATIVE phrase, VI-13, VI-15
Native character set, III-14, VI-15
Native collating sequence, III-14, VI-15
Natural logarithm, A-49
Negated condition, VI-59
NEGATIVE, VI-58
Nested source programs, X-1, X-8
Nested statements, IV-40
NEXT GROUP clause, XIII-20, XIII-48
NEXT PAGE phrase
  LINE NUMBER clause, XIII-46
  NEXT GROUP clause, XIII-48
NEXT phrase
  Indexed I-O module, IX-28
  Relative I-O module, VIII-26
  Sequential I-O module, VII-44
NEXT SENTENCE phrase
  IF statement, VI-90
  SEARCH statement, VI-122
NO DATA phrase, XIV-23
NO REWIND phrase, VII-35, VII-39, XIII-63, XIII-70
Noncontiguous elementary item, VI-18, X-13
  Level-number 77, VI-25
Nonnumeric comparison, VI-55
Nonnumeric literal, IV-9
  Continuation, IV-42

NOT
  EVALUATE statement, VI-84
  Logical operator, VI-59
  Relational operator, VI-55
NOT AT END-OF-PAGE phrase, VII-52
NOT AT END phrase
  Implicit scope terminator, IV-27
  READ statement, VII-44
  RETURN statement, XI-14
NOT AT EOP phrase, VII-52
NOT INVALID KEY phrase
  DELETE statement, VIII-19, IX-21
  READ statement, VIII-26, IX-28
  REWRITE statement, VIII-30, IX-33
  START statement, VIII-33, IX-36
  WRITE statement, VIII-37, IX-41
NOT ON EXCEPTION phrase, X-27
NOT ON OVERFLOW phrase
  STRING statement, VI-131
  UNSTRING statement, VI-136
NOT ON SIZE ERROR phrase
  ADD statement, VI-73
  COMPUTE statement, VI-76
  DIVIDE statement, VI-80
  MULTIPLY statement, VI-107
  SUBTRACT statement, VI-134
Notation rules, IV-1
Nucleus, VI-1
  Element summary, I-11
NUMERIC
  Class condition, VI-57
  INITIALIZE statements, VI-92
  Numeric category, IV-15, VI-29, VI-48, VI-104
Numeric character, III-15
Numeric class, IV-15, VI-56
Numeric comparison, VI-55
Numeric data item, VI-30
NUMERIC-EDITED, VI-92
Numeric edited category, IV-15, VI-29, VI-48, VI-104
Numeric edited data item, VI-30
Numeric function, A-8, A-10, A-29
Numeric literal, IV-10, VI-104
NUMVAL function, A-58
NUMVAL-C function, A-59

OBJECT-COMPUTER paragraph, VI-11, XVI-4
Object time switch, XV-2
Obsolete language element, I-7
Obsolete language element list, XVII-81
Occurrence number, IV-21, VI-124, VI-128
OCCURS clause, VI-26
  CORRESPONDING phrase, VI-68
  INITIALIZE statement, VI-92
  MOVE statement, VI-104
  REDEFINES clause, VI-38
  RENAMES clause, VI-40
  SEARCH statement, VI-123
  SET statement, VI-128
  SYNCHRONIZED clause, VI-45
  VALUE clause, VI-50
OF, IV-19, IV-20, XII-2
OFF, VI-127
OFF STATUS phrase, VI-13
ON, VI-127
ON EXCEPTION phrase, X-27
ON OVERFLOW phrase
  CALL statement, X-27
  STRING statement, VI-131
  UNSTRING statement, VI-136
ON phrase, VII-17, VIII-13, IX-15
ON SIZE ERROR phrase, VI-67
  ADD statement, VI-73
  COMPUTE statement, VI-76
  DIVIDE statement, VI-80
  MULTIPLY statement, VI-107
  SUBTRACT statement, VI-134
Open mode, II-4, VII-39, VIII-21, IX-23, XIII-70
OPEN statement
  Imperative statement, IV-39
  Indexed I-O module, IX-23
  Relative I-O module, VIII-21
  Report Writer module, XIII-70
  Sequential I-O module, VII-39
Operational sign, IV-16
Operator
  Arithmetic, IV-9, VI-52
  Logical, VI-59
  Relational, VI-54, VI-55
Optional modules, I-6
OPTIONAL phrase
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Sequential I-O module, VII-7
Optional word, IV-9
OR
  Abbreviated combined relation condition, VI-61
  Combined condition, VI-59
  Evaluation order, VI-61
  Logical operator, VI-59
OR phrase, VI-136
ORD function, A-60
ORD-MAX function, A-61
ORD-MIN function, A-62
Ordinal function, A-60
ORGANIZATION IS INDEXED clause, IX-13
ORGANIZATION IS RELATIVE clause, VIII-12
ORGANIZATION IS SEQUENTIAL clause, VII-11, XIII-3
OTHER, VI-84

Out-of-line PERFORM statement, VI-111
OUTPUT phrase
  Communication description entry, XIV-4
  DISABLE statement, XIV-18
  ENABLE statement, XIV-20
  OPEN statement, VII-39, VIII-21, IX-23, XIII-70
  USE statement, VII-50, VIII-35, IX-39
OUTPUT PROCEDURE phrase, XI-8, XI-16
Overall language consideration, IV-1
OVERFLOW phrase
  CALL statement, X-27
  STRING statement, VI-131
  UNSTRING statement, VI-136
Overlapping operands, VI-69

'P' PICTURE symbol, VI-31
  USAGE clause, VI-46
PACKED-DECIMAL, VI-46
PADDING CHARACTER clause, VII-7, VII-12, XIII-3
PAGE
  SEND statement, XIV-26
  WRITE statement, VII-52
PAGE clause, XIII-11, XIII-17
PAGE-COUNTER, XIII-1, XIII-12
  Qualified, IV-19, IV-20
  Special register, IV-9
PAGE FOOTING (PF), XIII-55
Page footing presentation rules, XIII-37
PAGE HEADING (PH), XIII-55
Page heading group presentation rules, XIII-30
Paragraph, IV-35, IV-43
Paragraph header, III-17, IV-43
Paragraph-name, IV-6, IV-7, IV-35
  Qualified, IV-19
  Reference format, IV-43
Parentheses
  Arithmetic expression, VI-53
  Condition, VI-60
  PICTURE clause, VI-30
  Separators, IV-4
  Subscripting, IV-21
PERFORM statement, VI-109
  Conditional expression, VI-54
  Imperative statement, IV-39
  Segmentation, XVI-8
  SET statement, VI-128
  Transfer of control, IV-26
Period, IV-3
  DECIMAL-POINT IS COMMA clause, VI-17
  PICTURE symbol, VI-32, VI-33
  Separator, IV-4
PF, XIII-55
PH, XIII-55
Phrase, III-17
Physical record, IV-13, VII-23
PICTURE character-string, IV-12
PICTURE (PIC) clause, VI-29
  BLANK WHEN ZERO clause, VI-22
  CURRENCY SIGN clause, VI-17
  DECIMAL POINT IS COMMA clause, VI-17
  Linkage Section, X-13
  Report group description entry, XIII-21
  SIGN clause, VI-42
  STRING statement, VI-131
  SYNCHRONIZED clause, VI-44
  UNSTRING statement, VI-136
  USAGE clause, VI-46
  Working-Storage Section, VI-18
PLUS phrase
  LINE NUMBER clause, XIII-46
  NEXT GROUP clause, XIII-48
Plus (+) PICTURE symbol, VI-32, VI-34, VI-35
POINTER phrase
  STRING statement, VI-131
  UNSTRING statement, VI-136
POSITION phrase, VII-16
POSITIVE, VI-58
Precedence rules for PICTURE character-string, VI-36
PRESENT-VALUE function, A-63
Presentation rules tables, XIII-24
Prime record key, IX-14
Procedure, IV-35
Procedure branching statement, III-17, IV-26
Procedure Division, IV-35
  Communication module, XIV-17
  Debug module, XV-4
  Element summary, I-53
  Indexed I-O module, IX-18
  Inter-Program Communication module, X-25
  Nucleus, VI-51
  Relative I-O module, VIII-16
Procedure Division (continued)
  Report Writer module, XIII-62
  Segmentation module, XVI-6
  Sequential I-O module, VII-34
  Sort-Merge module, XI-8
Procedure Division header, IV-36, X-25
Procedure-name, III-18
  GO TO statement, VI-89
  MERGE statement, XI-8
  PERFORM statement, VI-109
  SORT statement, XI-16
  USE FOR DEBUGGING statement, XV-5
PROGRAM-ID paragraph
  Inter-Program Communication module, X-12
  Nucleus, VI-7
Program-name, III-18, IV-6
  Conventions, X-5
  End program header, X-11
  PROGRAM-ID paragraph, VI-7, X-12
Program classes, II-22
PROGRAM COLLATING SEQUENCE clause, VI-11
  ALPHABET clause, VI-15
Program run unit organization, II-18
Program segments, XVI-1

Pseudo-text, XII-2, XII-6
Pseudo-text delimiters, III-18, IV-5, IV-43
Pseudo-text format, IV-43
Punctuation characters, III-18
  Format punctuation, IV-2
  Separators, IV-4
PURGE statement, XIV-22

Qualification, IV-18
  Restriction, VI-1
Queue, III-18
Queue concepts, II-32
Queue hierarchy, II-33
Quotation mark
  Separator, IV-5
QUOTE, QUOTES, IV-11

Random access, II-4, VIII-2, IX-2
RANDOM function, A-64
RANGE function, A-65
RD entry, XIII-10, XIII-11
RD level indicator, III-13, X-22, XIII-11
  Reference format, IV-43
READ statement
  Indexed I-O module, IX-28
  Relative I-O module, VIII-26
  Sequential I-O module, VII-44
RECEIVE statement, XIV-23
Record
  Concepts, IV-14
  Fixed length records, II-3
  Implementor-defined record types, II-3
  Linkage records, X-14
  Logical, II-2, IV-13
  Physical, IV-13
  Variable length records, II-3
  Working storage records, VI-18
RECORD clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-30
  Sort-Merge module, XI-7
RECORD CONTAINS clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-30
  Sort-Merge module, XI-7
RECORD DELIMITER clause
  Report Writer module, XIII-3
  Sequential I-O module, VII-7, VII-13
Record description entry, III-19, IV-33, IV-34
Record description structure, VI-19, VII-21, XI-6, XIV-2
RECORD KEY clause, IX-14
Record-name, III-19, IV-6
  Conventions, X-6
RECORD VARYING clause, VII-22, VII-30, VIII-14, IX-16, XI-7
RECORDS, VII-23
RECORDS phrase, VII-17, VII-18
REDEFINES clause, VI-38
  CORRESPONDING phrase, VI-68
  INITIALIZE statement, VI-93
  Procedure Division header, X-25
  SYNCHRONIZED clause, VI-44
  VALUE clause, VI-49
REEL, VII-35, XIII-63
Reference format, IV-41
  Restriction, VI-2
  Text-words, XII-4
Reference modification, IV-22, A-12, B-6
  Restriction, VI-2
  STRING statement, VI-131
  UNSTRING statement, VI-136
  USE FOR DEBUGGING statement, XV-5
Relation character, IV-9
Relation condition, VI-54
  Abbreviated combined, VI-60
  Indexed data item, VI-46
  Nonnumeric operands, VI-55
  Numeric operands, VI-55
Relational operator, VI-54, VI-55
Relative file, VIII-1
Relative I-O module, VIII-1
  Element summary, I-22
RELATIVE KEY phrase, VIII-8, VIII-10
Relative organization, II-2, VIII-1, VIII-12
Relative record number, VIII-1, VIII-10
Relative subscripting, IV-22
RELEASE statement, XI-13
  Imperative statement, IV-39
REM function, A-66
REMAINDER phrase, VI-80
REMOVAL phrase, VII-35, XIII-63
RENAMES clause, VI-21, VI-40
  CORRESPONDING phrase, VI-68
  INITIALIZE statement, VI-92
  Level-number, IV-15, VI-25
  PICTURE clause, VI-29
REPLACE statement, XII-6
  Compiler directing statement, IV-38
REPLACING LINE phrase, XIV-26
REPLACING phrase
  COPY statement, XII-2
  INITIALIZE statement, VI-92
  INSPECT statement, VI-94
REPORT clause, XIII-9
Report description entry, III-20, X-22, XIII-10, XIII-11
Report file, XIII-1, XIII-3
REPORT FOOTING (RF), XIII-55
Report footing presentation rules, XIII-39
Report group description entry, III-20, XIII-10, XIII-20
REPORT HEADING (RH), XIII-55
Report heading group presentation rules, XIII-27

Report-name, III-21, IV-6, XIII-9, XIII-11, XIII-66, XIII-75
  Conventions, X-6
Report Section, IV-33, XIII-10
Report writer concepts, II-8
Report Writer module, XIII-1
  Element summary, I-33
Required words, IV-8
RERUN clause
  Indexed I-O module, IX-15
  Relative I-O module, VIII-13
  Sequential I-O module, VII-15, VII-17
RESERVE clause
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-7, VII-14
Reserved word, IV-8, IV-45
  Implementation, I-8, B-3
Reserved word list, IV-45
RESET phrase, XIII-52
RETURN statement, XI-14
  Conditional statement, IV-37
REVERSE function, A-67
REVERSED phrase, VII-39
REWRITE statement
  Indexed I-O module, IX-33
  Relative I-O module, VIII-30
  Sequential I-O module, VII-48
RF, XIII-55
RH, XIII-55
RIGHT, VI-24, VI-44
ROUNDED phrase, VI-67
  ADD statement, VI-73
  COMPUTE statement, VI-76
  DIVIDE statement, VI-80
  MULTIPLY statement, VI-107
  SUBTRACT statement, VI-134
Routine-name, III-21, IV-6, VI-83
Rules, IV-3
RUN, VI-130

'S' PICTURE symbol, VI-31
  SIGN clause, VI-42
  USAGE clause, VI-46
SAME clause
  Indexed I-O module, IX-15
  Relative I-O module, VIII-13
  Report Writer module, XIII-5
  Sequential I-O module, VII-15, VII-19
SAME RECORD AREA clause
  Indexed I-O module, IX-15
  Relative I-O module, VIII-13
  Sequential I-O module, VII-15, VII-19
  Sort-Merge module, XI-3, XI-4
SAME SORT AREA clause, XI-3, XI-4
SAME SORT-MERGE AREA clause, XI-3, XI-4
Scope of names, X-4
Scope of statements, IV-40
Scope terminators, IV-27, IV-40
SD level indicator, III-13, XI-6, XI-7
  Reference format, IV-43
SEARCH statement, VI-122
  Conditional expression, VI-54
  SET statement, VI-128
  USAGE IS INDEX clause, VI-46
Section, IV-35, IV-43
Section header, III-22, IV-43
Section-name, IV-6, IV-7
SECURITY paragraph, VI-6
Segment, XVI-1
SEGMENT-LIMIT clause, XVI-5
Segment-number, III-22, IV-6, XVI-7
SEGMENT phrase, XIV-23
Segmentation classification, XVI-2
Segmentation control, XVI-3
Segmentation module, XVI-1
  Element summary, I-39
SELECT clause
  Indexed I-O module, IX-8
  Relative I-O module, VIII-8
  Report Writer module, XIII-3
  Sequential I-O module, VII-7
  Sort-Merge module, XI-2
Semicolon, IV-2, IV-4
  Interchangeable with comma, IV-2
SEND statement, XIV-26
  Mnemonic-name, VI-13
  SPECIAL-NAMES paragraph, VI-13
  STOP statement, VI-130
Sentence, IV-35, IV-37, IV-43
SEPARATE CHARACTER phrase, VI-42
Separator, IV-4
  Restriction, VI-1
SEQUENCE clause, VI-11
Sequence number, IV-42
  Sequence number area, IV-41
Sequential access, II-3, VII-1, VIII-2, IX-2
Sequential file, VII-1
Sequential I-O module, VII-1
  Element summary, I-19
Sequential organization, II-1, VII-1, VII-11
SET statement, VI-127
  Imperative statement, IV-39
  SPECIAL-NAMES paragraph, VI-15
Shared memory area, II-17
Sharing data, X-4
Sharing files, X-4
SIGN clause, VI-42
  Class condition, VI-57
  MOVE statement, VI-105
  Operational sign, IV-16
  PICTURE clause, VI-31
  Report group description entry, XIII-21, XIII-49
Sign condition, VI-58
Simple condition, VI-54
Simple insertion editing, VI-33
SIN function, A-68
Single character substitution, IV-4

SIZE, VI-11, VI-131
SIZE ERROR phrase, VI-67
  ADD statement, VI-73
  COMPUTE statement, VI-76
  DIVIDE statement, VI-80
  MULTIPLY statement, VI-107
  SUBTRACT statement, VI-134
Slant (/) comment line, IV-42
Sort file, XI-1
Sort-merge file description entry, III-23, XI-6, XI-7
Sort-Merge module, XI-1
  Element summary, I-30
SORT statement, XI-16
  Imperative statement, IV-39
  Segmentation, XVI-9
  Transfer of control, IV-26
Sorting, II-5
SOURCE clause, XIII-21, XIII-51
SOURCE-COMPUTER paragraph, VI-10
  WITH DEBUGGING MODE phrase, XV-3
Source program, IV-29, VI-3
  COPY statement, XII-2
  Separately compiled, VI-4
Source text manipulation module, XII-1
  Element summary, I-32
Space, IV-4
SPACE/SPACES, IV-11, VI-104
Special character, III-23, IV-3
Special-character words, IV-3, IV-9
Special insertion editing, VI-33
SPECIAL-NAMES paragraph, VI-13
  ACCEPT statement, VI-71
  Condition-name, IV-7
  DISPLAY statement, VI-78
  Mnemonic-name, IV-7
  SEND statement, XIV-26
  Switch-status condition, VI-58
  WRITE statement, VII-52
Special registers, IV-9
  DEBUG-ITEM, XV-1
  LINAGE-COUNTER, VII-5
  LINE-COUNTER, XIII-1
  PAGE-COUNTER, XIII-1
SQRT function, A-69
Standard alignment rules, IV-16
Standard COBOL, I-6
Standard data format, III-24, IV-13
STANDARD-DEVIATION function, A-70
Standard language element acceptance, I-7
STANDARD-1 phrase
  ALPHABET clause, VI-13, VI-15
  RECORD DELIMITER clause, VII-13
STANDARD-2 phrase, VI-13, VI-15
Standardization of COBOL, XVII-11
START statement
  Indexed I-O module, IX-36
  Relative I-O module, VIII-33
Statement, IV-35, IV-37
STATUS KEY clause, XIV-3, XIV-4
STOP statement, VI-130
  Figurative constant, IV-11
  Imperative statement, IV-39
  Transfer of control, IV-26
STRING statement, VI-131
  Figurative constant, IV-11
  Imperative statement, IV-39
Subscript, A-9
Subscripted identifier, A-11
Subscripting, IV-21
  Concepts, II-14
  Condition-name, IV-24
  Conditional variable, IV-24
  MOVE statement, VI-103
  Qualified, IV-21
  Report Section, XIII-2
  Restriction, VI-2
  SET statement, VI-128
Subsets of Standard COBOL, I-5
Substantive changes, XVII-42, B-27
Substitute language elements, I-7
SUBTRACT CORRESPONDING (SUBTRACT CORR), VI-68, VI-134
SUBTRACT statement, VI-134
  Composite of operands, VI-69
  COMPUTE statement, VI-76
  Conditional statement, IV-37
  CORRESPONDING (CORR), VI-68
  Data conversion, VI-69
  Decimal alignment, VI-69
  Imperative statement, IV-39
  Maximum operand size, VI-69
  Multiple results, VI-69
SUM clause, XIII-21, XIII-52
SUM function, A-71
Summary of elements by COBOL division, I-40
Summary of elements by module, I-10
Summary of differences, XVII-16
SUPPRESS statement, XIII-74
  Imperative statement, IV-39
Switch-status condition, VI-58
Symbolic-character, III-24, IV-6
Symbolic-character figurative constant, IV-11
SYMBOLIC CHARACTERS clause, VI-13
SYMBOLIC DESTINATION clause, XIV-4
SYMBOLIC QUEUE clause, XIV-3
SYMBOLIC SOURCE clause, XIV-3
SYMBOLIC SUB-QUEUE-1 clause, XIV-3
SYMBOLIC SUB-QUEUE-2 clause, XIV-3
SYMBOLIC SUB-QUEUE-3 clause, XIV-3
SYMBOLIC TERMINAL clause, XIV-4
SYNCHRONIZED (SYNC) clause, VI-44
  Elementary data item, VI-21
  VALUE clause, VI-49
Syntax rules, IV-3
System-name, IV-8

Table, II-12, VI-122, VI-127
Table definition, II-12
Table element, II-12, VI-122, VI-127

Table handling, II-12, VI-127
TALLYING phrase
  INSPECT statement, VI-94
  UNSTRING statement, VI-136
TAN function, A-72
Tangent function, A-72
TERMINAL phrase, XIV-18, XIV-20
TERMINATE statement, XIII-75
TEST AFTER phrase, VI-109, VI-110
TEST BEFORE phrase, VI-109, VI-110
TEXT LENGTH clause, XIV-3, XIV-4
Text-name, III-25, IV-6, XII-2
  Qualified, IV-19
Text word, III-25, XII-2
THEN, VI-90
THROUGH (THRU)
  ALPHABET clause, VI-13
  EVALUATE statement, VI-84
  MERGE statement, XI-8
  PERFORM statement, VI-109
  RENAMES clause, VI-40
  SORT statement, XI-16
  VALUE clause, VI-48
TIME, VI-72
TIMES, VI-109
TRAILING, VI-42
Transaction communication, II-35
Transfer of control, II-23, IV-25
TRUE, VI-84, VI-127
TYPE clause, XIII-20, XIII-55

Unary arithmetic operator, VI-52
Unary minus, VI-52
Unary plus, VI-52
Undefined language element list, XVII-96, A-24, B-47
Uniqueness of reference, IV-17
UNIT, VII-35, XIII-63
UNSTRING statement, VI-136
  Figurative constant, IV-11
  Imperative statement, IV-39
UNTIL phrase, VI-109, VI-110
UP BY, VI-127
UPON phrase
  DISPLAY statement, VI-78
  SUM clause, XIII-52
UPPER-CASE function, A-73
Uppercase letters, IV-6, VI-29, VI-56
USAGE clause, VI-46
  Class condition, VI-57
  INSPECT statement, VI-95
  Relation condition, VI-54
  Report group description entry, XIII-20, XIII-21, XIII-60
  SIGN clause, VI-42
  STRING statement, VI-131
  SYNCHRONIZED clause, VI-44
  UNSTRING statement, VI-136
  VALUE clause, VI-49
USAGE IS INDEX clause, VI-46
  CORRESPONDING phrase, VI-68
  SEARCH statement, VI-123
  Working-Storage Section, VI-18
USE BEFORE REPORTING statement, X-35, XIII-78
USE FOR DEBUGGING statement, XV-5
USE statement
  Compiler directing statement, IV-38
  Declaratives, IV-35
  Indexed I-O module, IX-39
  Inter-Program Communication module, X-34
  Relative I-O module, VIII-35
  Report Writer module, XIII-76
  Sequential I-O module, VII-50
User-defined words, IV-6
  Restrictions, VI-1
USING phrase in CALL statement, X-27
  Index data item, VI-46
USING phrase in MERGE statement, XI-8
USING phrase in Procedure Division header, X-25
  Index data item, VI-46
USING phrase in SORT statement, XI-16

'V' PICTURE symbol, VI-32
  USAGE clause, VI-46
VALUE clause, VI-48
  Report Writer module, XIII-21, XIII-61
  SET statement, VI-129
VALUE OF clause
  Indexed I-O module, IX-16
  Relative I-O module, VIII-14
  Report Writer module, XIII-7
  Sequential I-O module, VII-22, VII-33
Variable length records, II-3, VII-30
Variable occurrence data item, III-26, VI-27, VI-50
VARIANCE function, A-74
VARYING IN SIZE phrase, VII-30
VARYING phrase
  PERFORM statement, VI-110
  SEARCH statement, VI-122
Verb, III-26

WHEN-COMPILED function, A-75
WHEN phrase
  EVALUATE statement, VI-84
  SEARCH statement, VI-122
WITH DATA phrase, XIV-23
WITH DEBUGGING MODE clause, VI-10, XV-2, XV-3
WITH DUPLICATES phrase
  ALTERNATE RECORD KEY clause, IX-8, IX-11
  SORT statement, XI-16
WITH EGI phrase, XIV-26
WITH EMI phrase, XIV-26
WITH ESI phrase, XIV-26
WITH FOOTING phrase, VII-27
WITH identifier phrase, XIV-26
WITH KEY phrase, XIV-18, XIV-20

WITH LOCK phrase, VII-35, VIII-17, IX-19, XIII-63
WITH NO ADVANCING phrase, VI-78
WITH NO REWIND phrase, VII-35, VII-39, XIII-63, XIII-70
WITH POINTER phrase
  STRING statement, VI-131
  UNSTRING statement, VI-136
WITH TEST phrase, VI-109
Word, IV-1, A-9
WORDS, VI-11
Working-Storage Section, IV-33, VI-18
  VALUE clause, VI-49
WRITE statement
  Conditional statement, IV-37
  Imperative statement, IV-39
  Indexed I-O module, IX-41
  Mnemonic-name, VI-13
  Relative I-O module, VIII-37
  Sequential I-O module, VII-52
  SPECIAL-NAMES paragraph, VI-13

'X' PICTURE symbol, VI-32

'Z' PICTURE symbol, VI-32, VI-35
ZERO/ZEROS/ZEROES, IV-11, VI-104
ZERO in sign condition, VI-58
Zero suppression editing, VI-35

'0' PICTURE symbol, VI-32, VI-33
'9' PICTURE symbol, VI-32, VI-46
'01' entry, IV-14, VI-25, XIII-20
'66' RENAMES data description entry, VI-21, VI-25, VI-40
'77' item description entry, VI-18, VI-25, X-13

'88' condition-name data description entry, VI-21, VI-25

> relation, VI-54
< relation, VI-54
= relation, VI-54
+ operator, VI-52
+ PICTURE symbol, VI-32, VI-34, VI-35
— continuation line, IV-42
— operator, VI-52
— PICTURE symbol, VI-32, VI-34, VI-35
* comment line, IV-42
* operator, VI-52
* PICTURE symbol, VI-32, VI-35
/ comment line, IV-42
/ operator, VI-52
/ PICTURE symbol, VI-32, VI-33
** operator, VI-52
== pseudo-text delimiter, III-18, IV-42